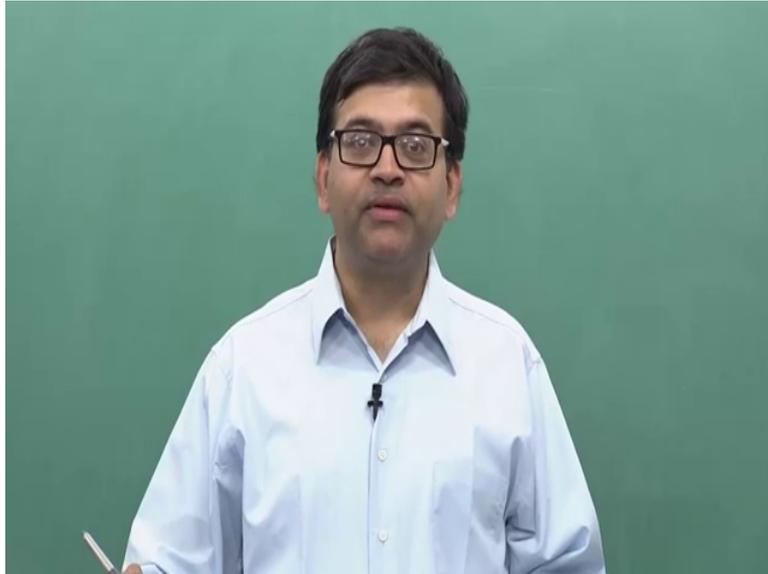


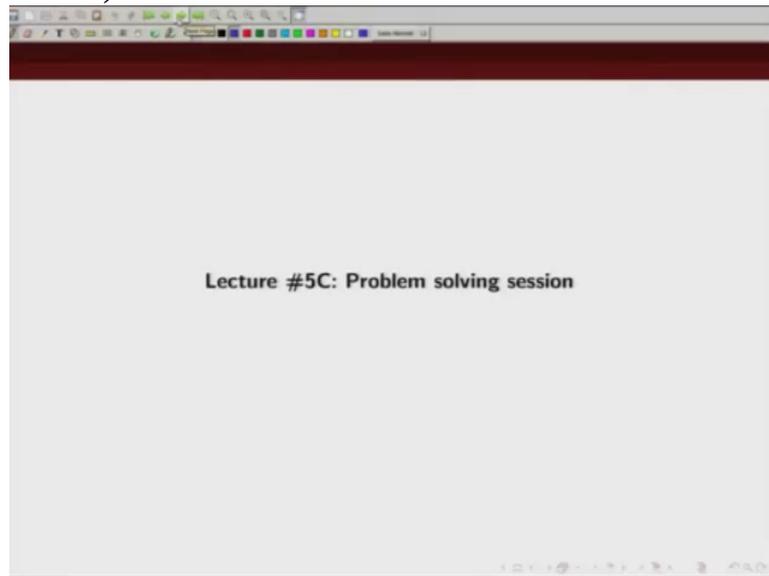
An Introduction to Coding Theory
Professor Adrish Banerji
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
Module 02
Lecture Number 10
Problem Solving Session-II

(Refer Slide Time 00:27)



So, so far we have studied what are linear block codes, how do we describe linear block codes using generator matrix and parity check matrix, we talked about how we can use error correcting codes for error detection and error correction and we discussed the distance properties of linear block codes. Today we will spend some time solving some problems on linear block codes so today's session

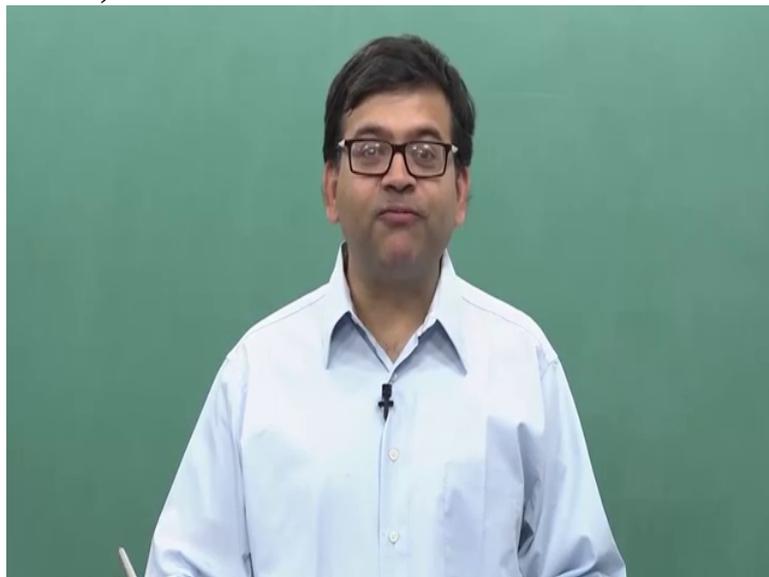
(Refer Slide Time 00:44)



will be on problem solving.

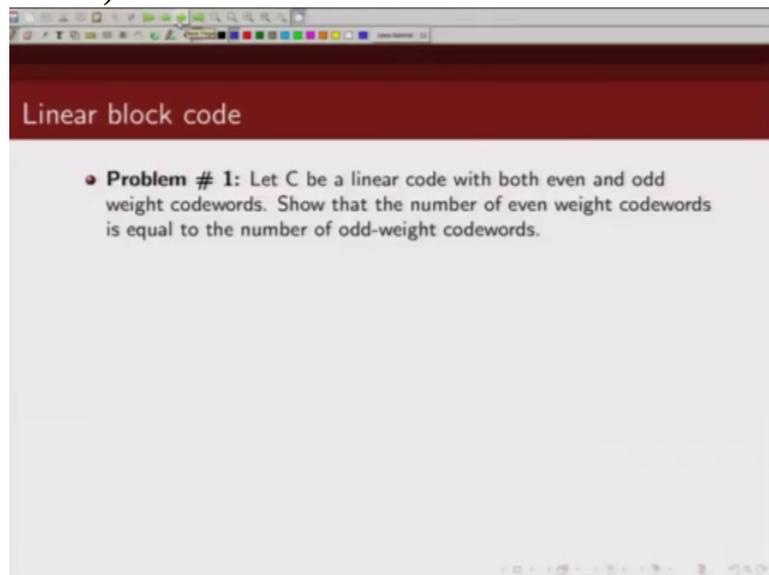
So the first problem

(Refer Slide Time 00:47)



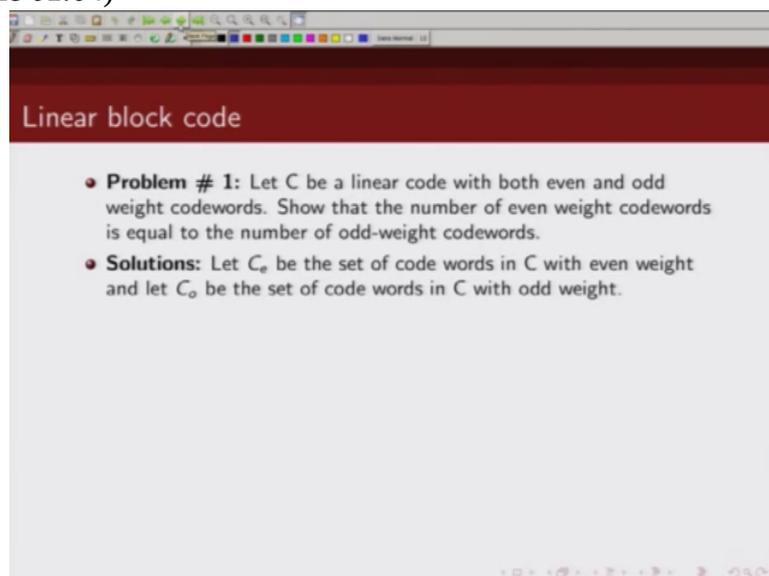
that we will look at is let C be a linear

(Refer Slide Time 00:51)



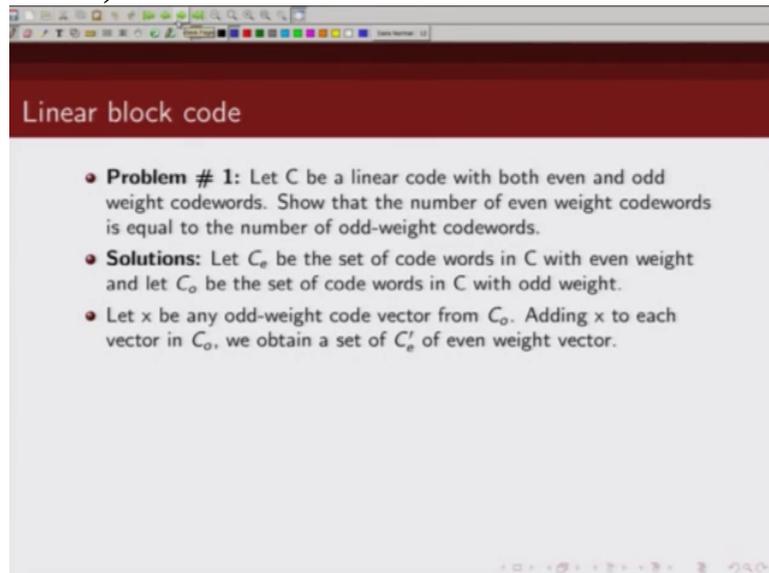
code with both even and odd weight codewords. Prove that the number of even bit codewords is equal to number of odd bit codewords.

(Refer Slide Time 01:04)



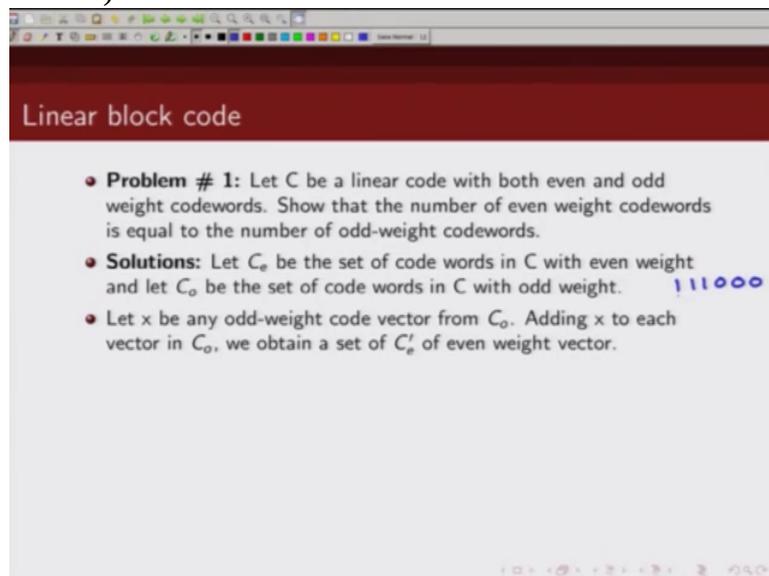
So let us denote the set of even codewords in C by $C_{sub\ e}$ and set of odd codewords in C by $C_{sub\ o}$.

(Refer Slide Time 01:20)



Now let us consider an odd weight codeword x which is taken from this set C_o and let us add x to each of the codewords which are there in the set C_o . So if we add a odd weight codeword to another odd weight codeword what we will get is a even weight codeword. For example let's say I add 1 1 1 0 0 0 and I add

(Refer Slide Time 01:58)



1 0 1 0 1 0 so

(Refer Slide Time 02:04)

Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C_e of even weight vector.

Handwritten blue text: 111000
 101010

 010010

this first codeword this is odd weight codeword its weight is 3. Similarly this codeword also has weight 3. If I add both of them what do I get? I get 0 1 0 0 1 0 and this a

(Refer Slide Time 02:26)

Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C_e of even weight vector.

Handwritten blue text: 111000
 101010

 010010

even weight codeword. So when I add x which is an odd weight code vector and I add x to each of the elements in this set C_o what I get is a set of even codeword vectors and let us denote that set by

(Refer Slide Time 02:49)

Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.

Handwritten example:
$$\begin{array}{r} 111000 \\ 101010 \\ \hline 010010 \end{array}$$

C_e prime. Now

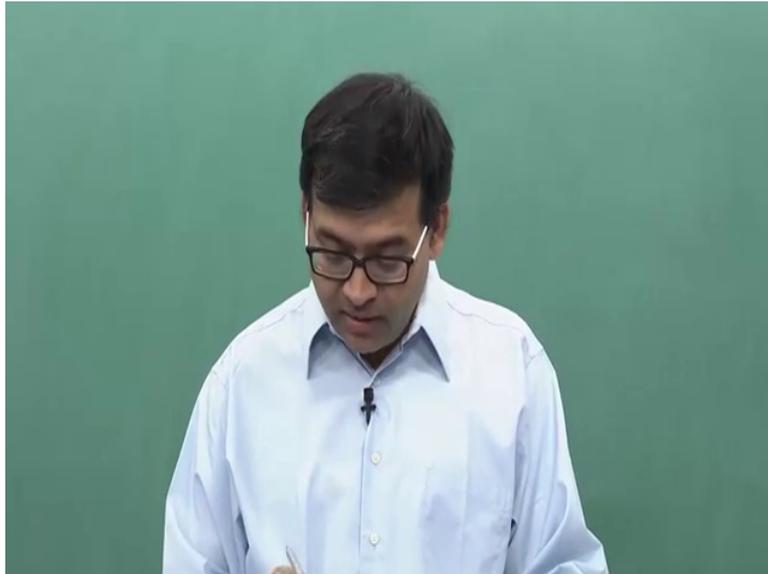
(Refer Slide Time 02:52)

Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.

the number of code vectors in C_e prime is going to be equal to number of vectors in C_o .
Why, because

(Refer Slide Time 03:02)



how did we get this C_e ? We added

(Refer Slide Time 03:05)

Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.

an odd vector x to the set C_o . So number of vectors in this set is going to be equal to

(Refer Slide Time 03:15)

Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.

number of vectors in C_o . Hence number of elements in

(Refer Slide Time 03:23)

Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.

C_e prime is going to be same as number of elements in

(Refer Slide Time 03:29)

The slide is titled "Linear block code" and contains the following text:

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.

in C_o and since we know that this set

(Refer Slide Time 03:35)

The slide is titled "Linear block code" and contains the following text:

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.

of even vectors C_e is the subset of

(Refer Slide Time 03:41)

The slide is titled "Linear block code" and contains the following text:

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.

set of even vectors we can write from this that number of elements in the

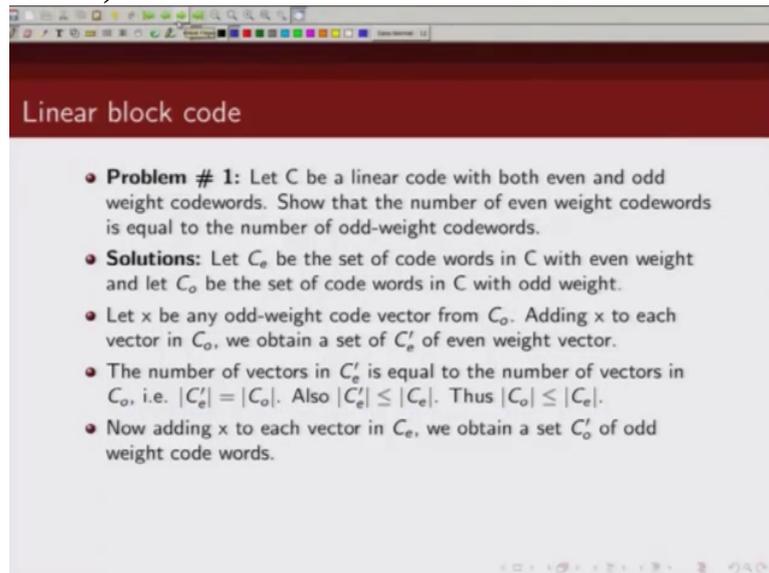
(Refer Slide Time 03:50)

The slide is titled "Linear block code" and contains the following text:

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.

set of number of odd codewords is going to be a subset of number of even codewords. Next,

(Refer Slide Time 04:02)

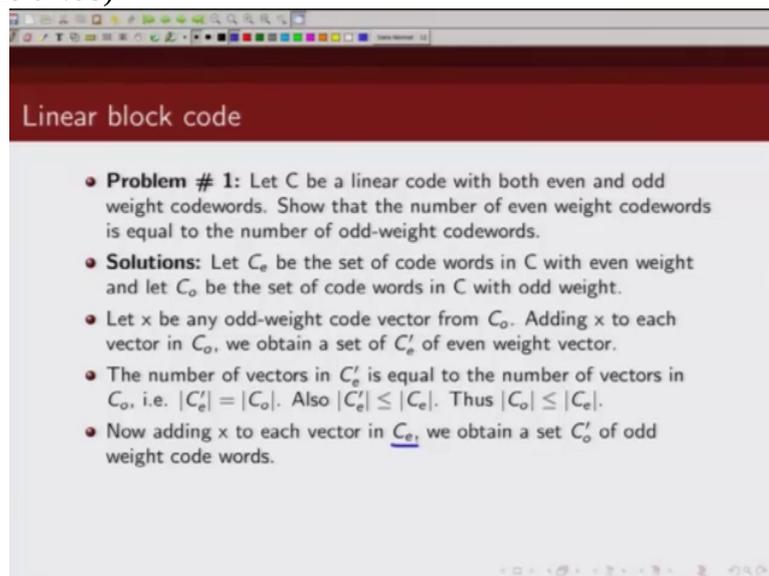


Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.
- Now adding x to each vector in C_e , we obtain a set C'_o of odd weight code words.

let us add the same odd weight codeword now to

(Refer Slide Time 04:08)



Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.
- Now adding x to each vector in C_e , we obtain a set C'_o of odd weight code words.

all the vectors in the set C_e . So if we add an odd weight codeword to set of even weight codewords what we will get is a set of odd codewords. So the

(Refer Slide Time 04:28)

Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.
- Now adding x to each vector in C_e , we obtain a set C'_o of odd weight code words.
- The number of vectors in C'_o is equal to the number of vectors in C_e and $|C'_o| \leq |C_o|$. Hence $|C_e| \leq |C_o|$.

number of vectors in C_o prime is going to be equal to number of vectors in, number of even vectors.

(Refer Slide Time 04:41)

Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.
- Now adding x to each vector in C_e , we obtain a set C'_o of odd weight code words.
- The number of vectors in C'_o is equal to the number of vectors in C_e and $|C'_o| \leq |C_o|$. Hence $|C_e| \leq |C_o|$.

Why, because this set was generated by adding an odd vector x to the set of even codewords. So we can then write that C_o prime is equal to this ok,

(Refer Slide Time 05:06)

Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.
- Now adding x to each vector in C_e , we obtain a set C'_o of odd weight code words.
- The number of vectors in C'_o is equal to the number of vectors in C_e and $|C'_o| \leq |C_o|$. Hence $|C_e| \leq |C_o|$.

Handwritten note: $|C_o'| = |C_e|$

the set of codewords here is same as set of codewords here.

(Refer Slide Time 05:10)

Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.
- Now adding x to each vector in C_e , we obtain a set C'_o of odd weight code words.
- The number of vectors in C'_o is equal to the number of vectors in C_e and $|C'_o| \leq |C_o|$. Hence $|C_e| \leq |C_o|$.

Handwritten note: $|C_o'| = |C_e|$

Now we know that C_o is the subset of set of odd codewords. So then from this relation and this relation we can write that set of even codewords is the subset of set of number of elements in this is subset of number of is basically less than number of elements in this set. Now from this relation and this relation

(Refer Slide Time 05:43)

Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.
- Now adding x to each vector in C_e , we obtain a set C'_o of odd weight code words.
- The number of vectors in C'_o is equal to the number of vectors in C_e and $|C'_o| \leq |C_o|$. Hence $|C_e| \leq |C_o|$.

Handwritten notes: $|C'_o| = |C_e|$

both of them can be true only if number of elements in C_o is same as number of elements in C_e . So this relation we call it 1 and let's call it 2. These

(Refer Slide Time 05:58)

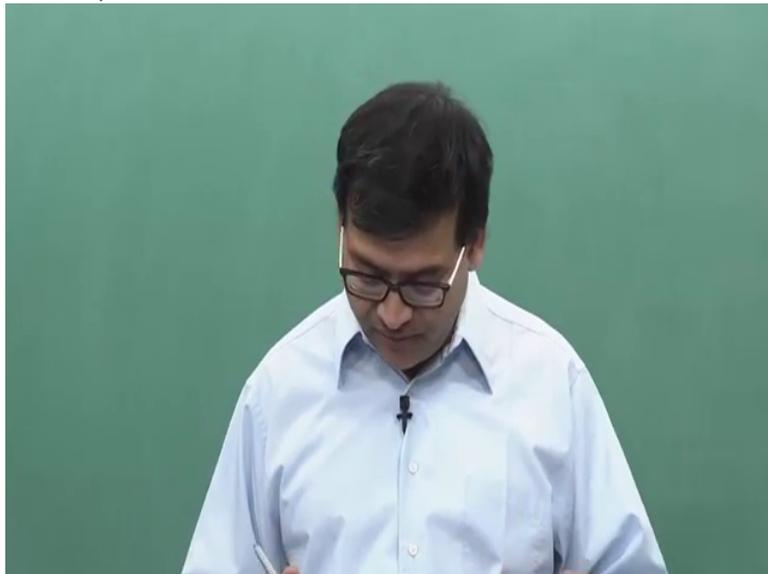
Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$. -①
- Now adding x to each vector in C_e , we obtain a set C'_o of odd weight code words.
- The number of vectors in C'_o is equal to the number of vectors in C_e and $|C'_o| \leq |C_o|$. Hence $|C_e| \leq |C_o|$. -②

Handwritten notes: $|C'_o| = |C_e|$

two relations are satisfied only if

(Refer Slide Time 06:02)



we have set of even codewords to be same as set of odd codewords.

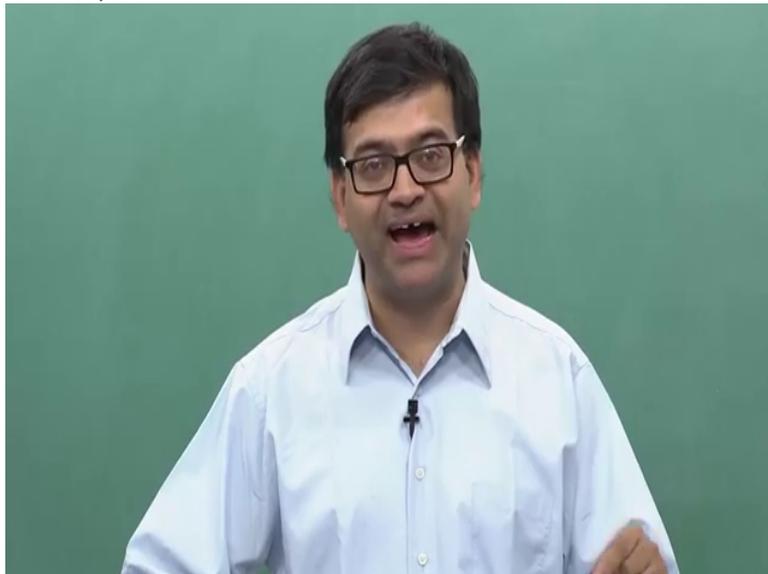
(Refer Slide Time 06:09)

Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$. -①
- Now adding x to each vector in C_e , we obtain a set C'_o of odd weight code words.
- The number of vectors in C'_o is equal to the number of vectors in C_e and $|C'_o| \leq |C_o|$. Hence $|C_e| \leq |C_o|$. -② $|C_e| = |C_o|$

Hence we prove that in a linear code with both even and odd codewords the number of even weight codewords is same as number

(Refer Slide Time 06:21)



of odd weight codewords.

(Refer Slide Time 06:24)

Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.
- Now adding x to each vector in C_e , we obtain a set C'_o of odd weight code words.
- The number of vectors in C'_o is equal to the number of vectors in C_e and $|C'_o| \leq |C_o|$. Hence $|C_e| \leq |C_o|$.
- Both these conditions are true only when $|C_e| = |C_o|$

So I repeat, this condition and this condition

(Refer Slide Time 06:29)

Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.
- Now adding x to each vector in C_e , we obtain a set C'_o of odd weight code words.
- The number of vectors in C'_o is equal to the number of vectors in C_e and $|C'_o| \leq |C_o|$. Hence $|C_e| \leq |C_o|$.
- Both these conditions are true only when $|C_e| = |C_o|$

will be simultaneously satisfied only when

(Refer Slide Time 06:35)

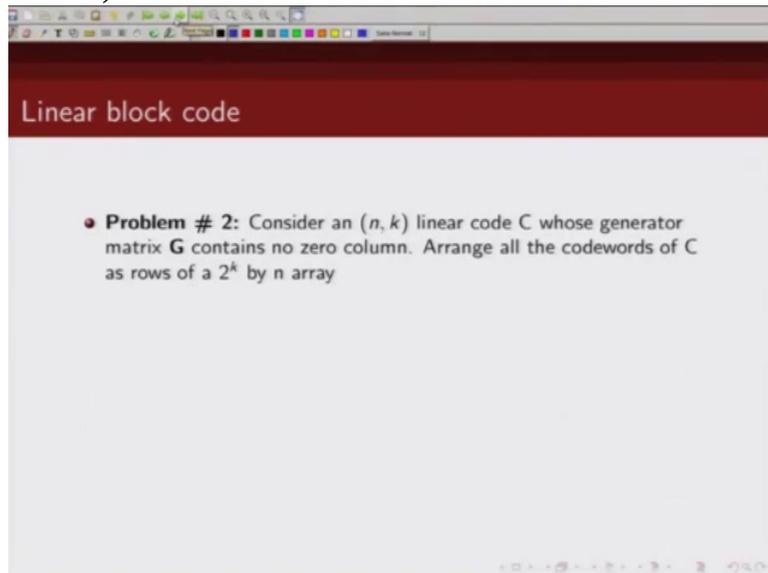
Linear block code

- **Problem # 1:** Let C be a linear code with both even and odd weight codewords. Show that the number of even weight codewords is equal to the number of odd-weight codewords.
- **Solutions:** Let C_e be the set of code words in C with even weight and let C_o be the set of code words in C with odd weight.
- Let x be any odd-weight code vector from C_o . Adding x to each vector in C_o , we obtain a set of C'_e of even weight vector.
- The number of vectors in C'_e is equal to the number of vectors in C_o , i.e. $|C'_e| = |C_o|$. Also $|C'_e| \leq |C_e|$. Thus $|C_o| \leq |C_e|$.
- Now adding x to each vector in C_e , we obtain a set C'_o of odd weight code words.
- The number of vectors in C'_o is equal to the number of vectors in C_e and $|C'_o| \leq |C_o|$. Hence $|C_e| \leq |C_o|$.
- Both these conditions are true only when $|C_e| = |C_o|$

the set of number of even codewords is same as set of number of odd codewords. And this proves our result.

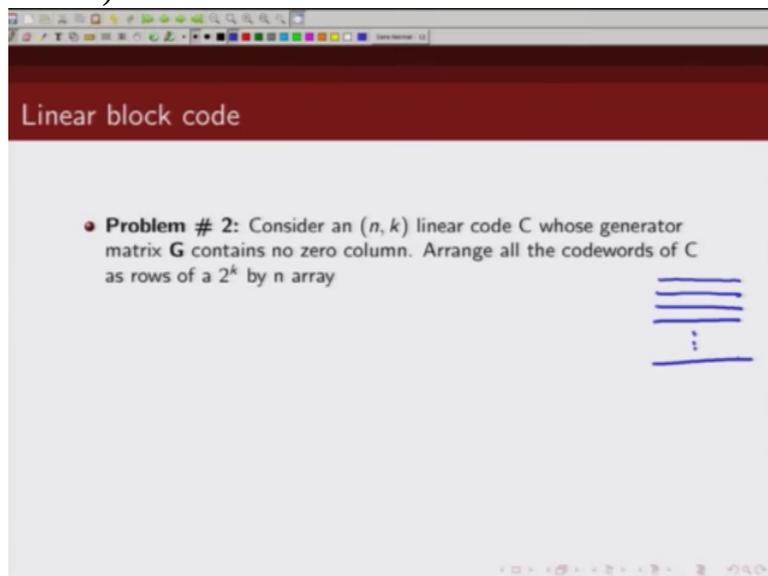
The next problem

(Refer Slide Time 06:48)



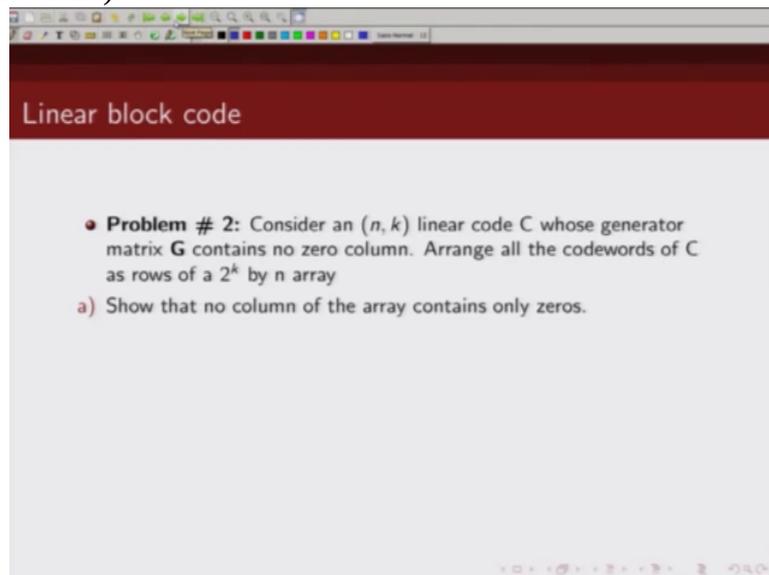
we will look out is as follows. Let us consider a linear n k code C whose generator matrix contains no zero column; now arrange all codewords of this linear code as rows of two raised to power k by n array. So what we are doing is we are arranging the 2^k codewords like this, in an array. So this array has dimensions 2^k cross n because total number of codewords

(Refer Slide Time 07:28)



are 2 raised to power k for a n k binary code and they are all n -bit. The first result that we are

(Refer Slide Time 07:37)

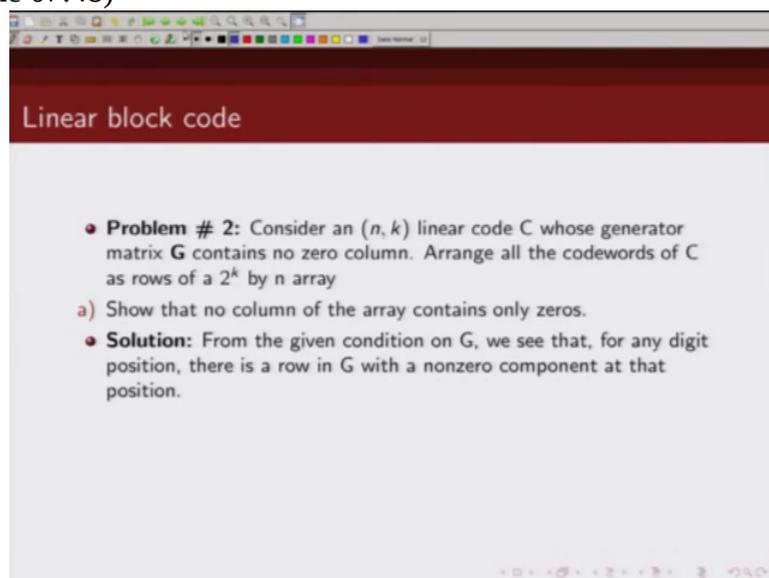


Linear block code

- **Problem # 2:** Consider an (n, k) linear code C whose generator matrix \mathbf{G} contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- a) Show that no column of the array contains only zeros.

going to show is no columns of this array contain 0. Now please note that we have

(Refer Slide Time 07:48)

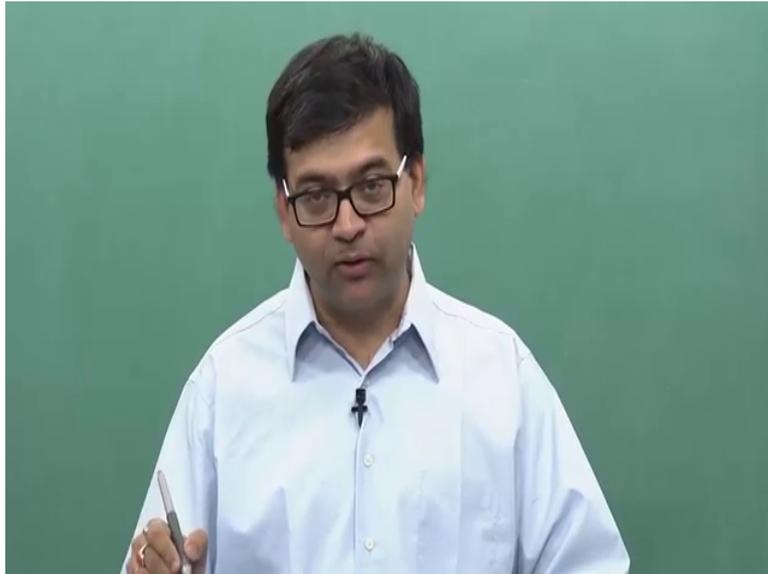


Linear block code

- **Problem # 2:** Consider an (n, k) linear code C whose generator matrix \mathbf{G} contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- a) Show that no column of the array contains only zeros.
- **Solution:** From the given condition on \mathbf{G} , we see that, for any digit position, there is a row in \mathbf{G} with a nonzero component at that position.

been given that this generator \mathbf{G} does not contain any zero column ok so from the given condition on \mathbf{G} we can see that for any position of any bit position there is a row in \mathbf{G} which has non-zero component at that particular

(Refer Slide Time 08:12)



bit location and

(Refer Slide Time 08:14)

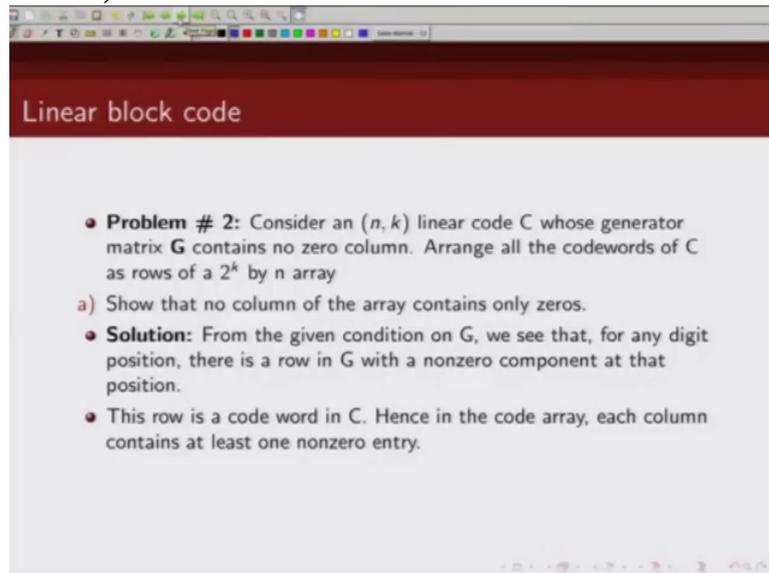
A slide titled "Linear block code" with a dark red header. The slide contains a problem and a solution. The problem asks to show that no column of a 2^4 by n array of codewords contains only zeros. The solution states that for any digit position, there is a row in G with a nonzero component at that position.

Linear block code

- **Problem # 2:** Consider an (n, k) linear code C whose generator matrix \mathbf{G} contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- a) Show that no column of the array contains only zeros.
- **Solution:** From the given condition on G , we see that, for any digit position, there is a row in G with a nonzero component at that position.

if this is

(Refer Slide Time 08:15)

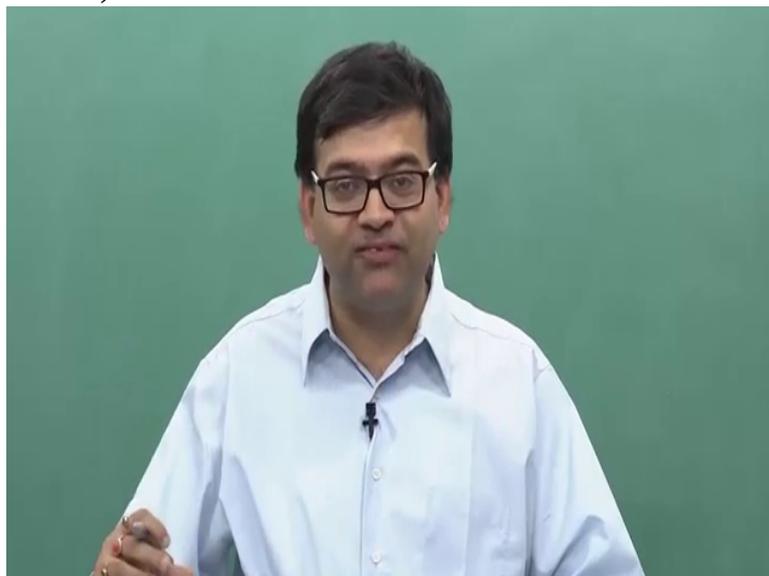


Linear block code

- **Problem # 2:** Consider an (n, k) linear code C whose generator matrix G contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- a) Show that no column of the array contains only zeros.
- **Solution:** From the given condition on G , we see that, for any digit position, there is a row in G with a nonzero component at that position.
- This row is a code word in C . Hence in the code array, each column contains at least one nonzero entry.

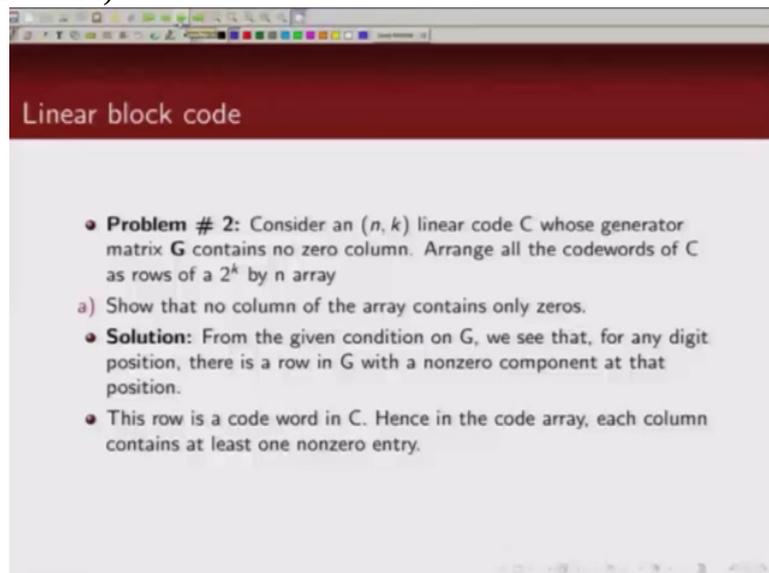
true what are the rows of, how do we generate the codewords? We generate the codewords by linear combination of these

(Refer Slide Time 08:22)



rows of the generator matrix. And since the generator matrix does not contain

(Refer Slide Time 08:29)



Linear block code

- **Problem # 2:** Consider an (n, k) linear code C whose generator matrix G contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- a) Show that no column of the array contains only zeros.
- **Solution:** From the given condition on G , we see that, for any digit position, there is a row in G with a nonzero component at that position.
- This row is a code word in C . Hence in the code array, each column contains at least one nonzero entry.

any zero column so each of these rows can be looked up as codeword in C so when we generate the codewords using this generator matrix in this code array, each column will have at least one

(Refer Slide Time 08:46)



non zero entry. So this follows from the fact

(Refer Slide Time 08:51)

The slide is titled "Linear block code" and contains the following text:

- **Problem # 2:** Consider an (n, k) linear code C whose generator matrix \mathbf{G} contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- a) Show that no column of the array contains only zeros.
- **Solution:** From the given condition on G , we see that, for any digit position, there is a row in G with a nonzero component at that position.
- This row is a code word in C . Hence in the code array, each column contains at least one nonzero entry.

that our generator matrix G does contain any zero column

(Refer Slide Time 08:59)

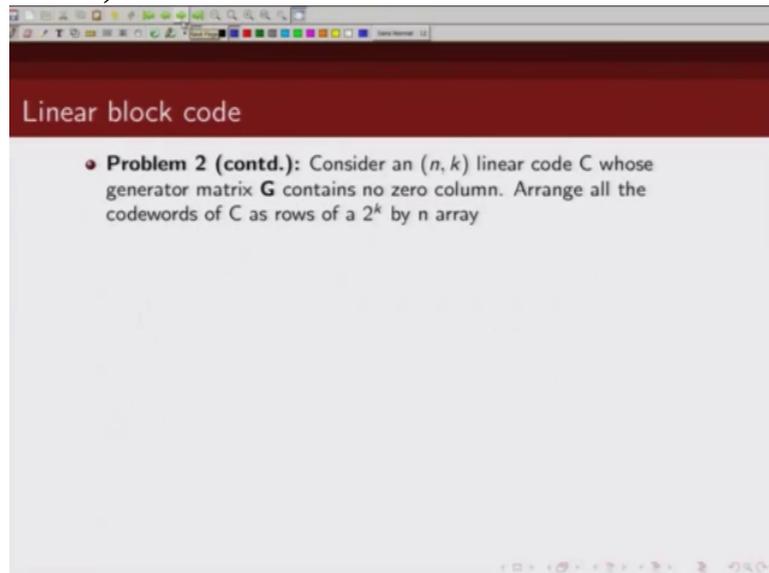
The slide is titled "Linear block code" and contains the following text:

- **Problem # 2:** Consider an (n, k) linear code C whose generator matrix \mathbf{G} contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- a) Show that no column of the array contains only zeros.
- **Solution:** From the given condition on G , we see that, for any digit position, there is a row in G with a nonzero component at that position.
- This row is a code word in C . Hence in the code array, each column contains at least one nonzero entry.
- Therefore no column in the code array contains only zeros.

and hence no column in this code array will have zeroes.

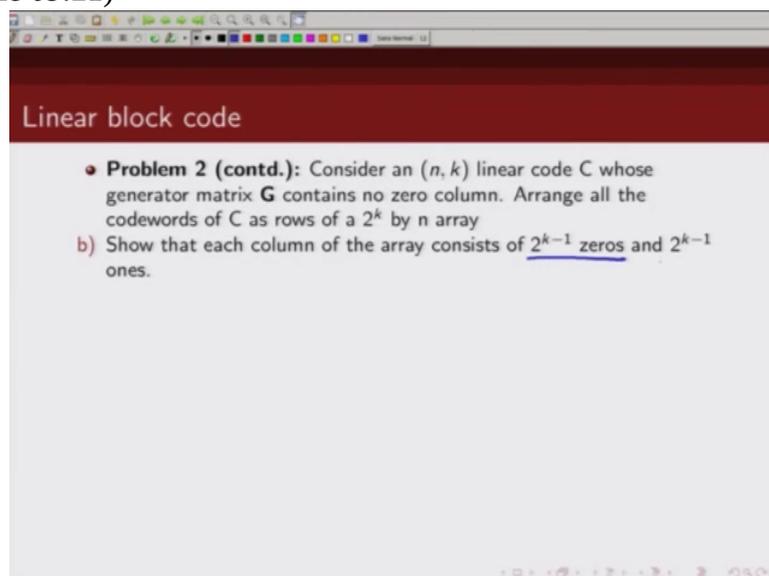
The next result that we

(Refer Slide Time 09:08)



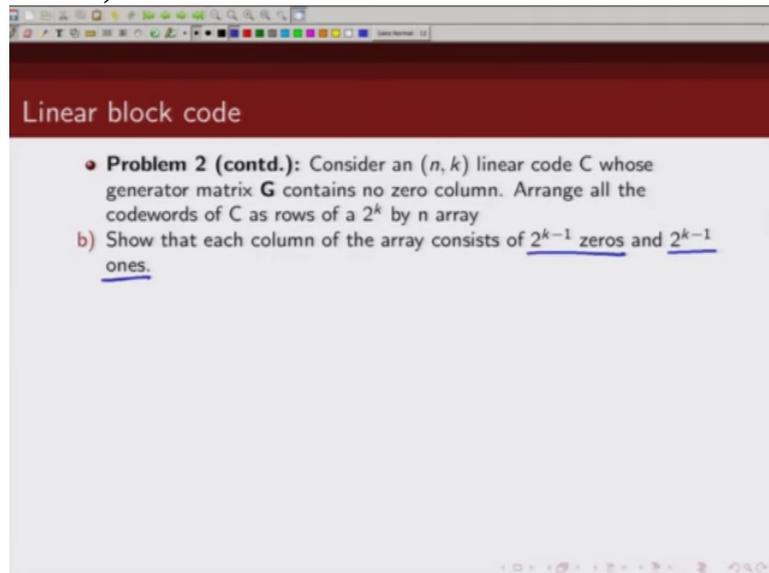
are going to show is in this array, in this 2^k by n array each column consists of equal numbers of

(Refer Slide Time 09:21)



0s and 1s.

(Refer Slide Time 09:25)

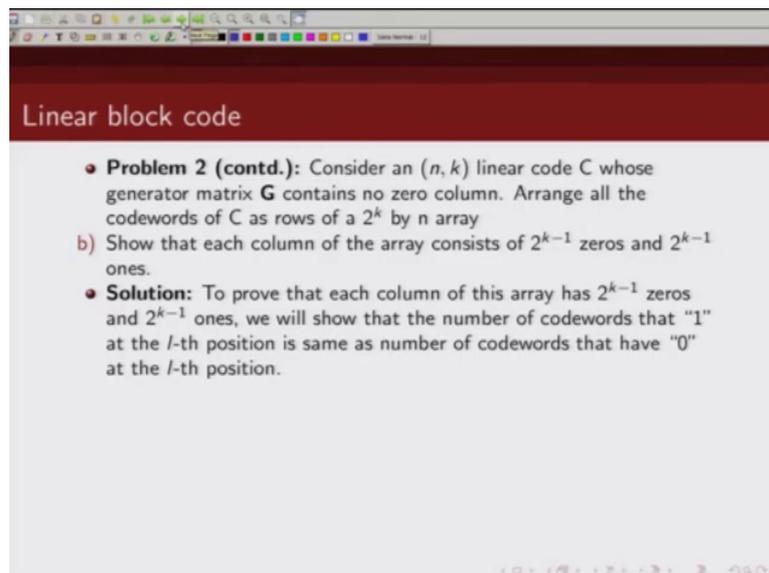


Linear block code

- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix \mathbf{G} contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.

There are total 2^k rows and n columns. So to prove

(Refer Slide Time 09:36)

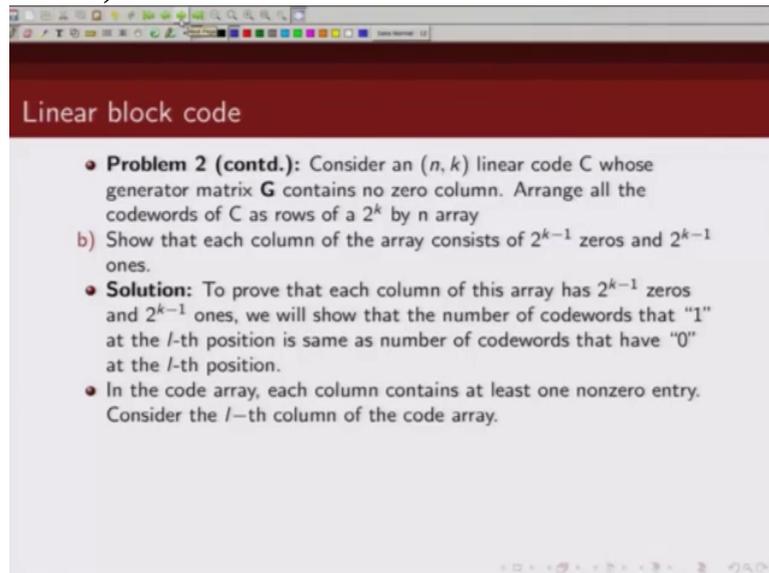


Linear block code

- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix \mathbf{G} contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.
- **Solution:** To prove that each column of this array has 2^{k-1} zeros and 2^{k-1} ones, we will show that the number of codewords that "1" at the l -th position is same as number of codewords that have "0" at the l -th position.

this what we will do is we will show that number of codewords that have 1 at l th location is same as number of codewords that have 0 at l th location. And in this way we will prove that this array has same number of 0s and 1s. So in this code array

(Refer Slide Time 10:01)

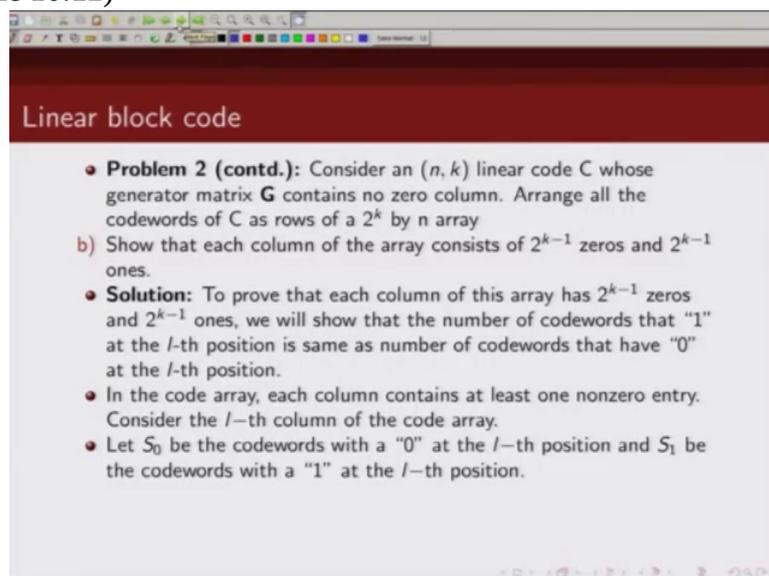


Linear block code

- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix \mathbf{G} contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.
- **Solution:** To prove that each column of this array has 2^{k-1} zeros and 2^{k-1} ones, we will show that the number of codewords that "1" at the l -th position is same as number of codewords that have "0" at the l -th position.
- In the code array, each column contains at least one nonzero entry. Consider the l -th column of the code array.

we know that each column will have at least one non-zero entry that we proved in the earlier result. So consider the l th column of this code array.

(Refer Slide Time 10:12)

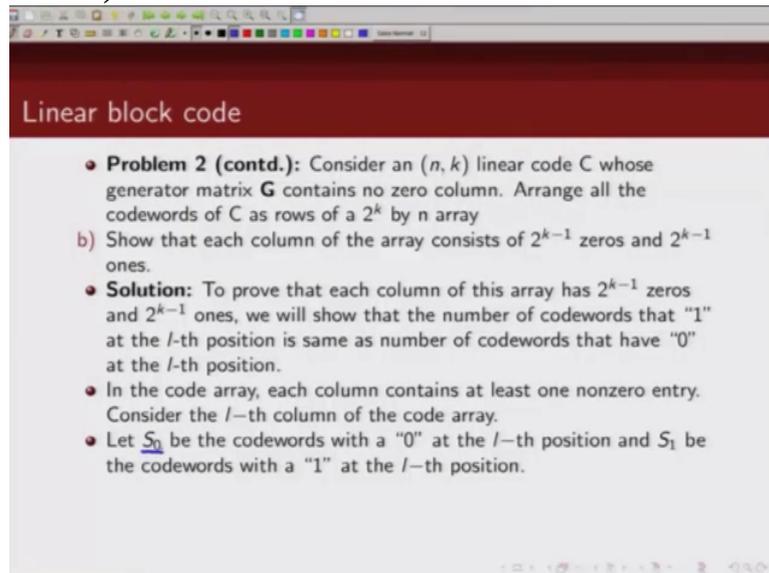


Linear block code

- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix \mathbf{G} contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.
- **Solution:** To prove that each column of this array has 2^{k-1} zeros and 2^{k-1} ones, we will show that the number of codewords that "1" at the l -th position is same as number of codewords that have "0" at the l -th position.
- In the code array, each column contains at least one nonzero entry. Consider the l -th column of the code array.
- Let S_0 be the codewords with a "0" at the l -th position and S_1 be the codewords with a "1" at the l -th position.

. Let us denote by S_0

(Refer Slide Time 10:17)

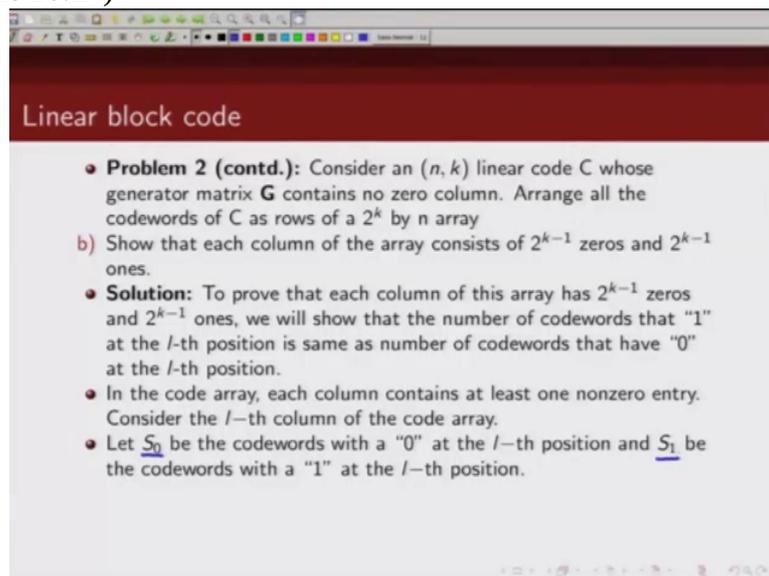


Linear block code

- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix \mathbf{G} contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.
- **Solution:** To prove that each column of this array has 2^{k-1} zeros and 2^{k-1} ones, we will show that the number of codewords that "1" at the l -th position is same as number of codewords that have "0" at the l -th position.
- In the code array, each column contains at least one nonzero entry. Consider the l -th column of the code array.
- Let S_0 be the codewords with a "0" at the l -th position and S_1 be the codewords with a "1" at the l -th position.

the set of codewords that has 0 at the l th location. And let us denote by S_1

(Refer Slide Time 10:27)



Linear block code

- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix \mathbf{G} contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.
- **Solution:** To prove that each column of this array has 2^{k-1} zeros and 2^{k-1} ones, we will show that the number of codewords that "1" at the l -th position is same as number of codewords that have "0" at the l -th position.
- In the code array, each column contains at least one nonzero entry. Consider the l -th column of the code array.
- Let S_0 be the codewords with a "0" at the l -th position and S_1 be the codewords with a "1" at the l -th position.

the set of codewords that have 1 at the l th location. Now we pick up a

(Refer Slide Time 10:36)

Linear block code

- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix G contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.
- **Solution:** To prove that each column of this array has 2^{k-1} zeros and 2^{k-1} ones, we will show that the number of codewords that "1" at the l -th position is same as number of codewords that have "0" at the l -th position.
- In the code array, each column contains at least one nonzero entry. Consider the l -th column of the code array.
- Let S_0 be the codewords with a "0" at the l -th position and S_1 be the codewords with a "1" at the l -th position.
- Let x be a codeword from S_1 . Adding x to each vector in S_0 , we obtain a set S'_1 of codewords with a "1" at the l -th position.

$$|S'_1| = |S_0| \text{ and } S'_1 \subseteq S_1$$

codeword x from the set S_1 that means x has 1 at l th location. Now if we add this codeword x to all the elements in the set S_0 what do we get?

(Refer Slide Time 10:58)

Linear block code

- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix G contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.
- **Solution:** To prove that each column of this array has 2^{k-1} zeros and 2^{k-1} ones, we will show that the number of codewords that "1" at the l -th position is same as number of codewords that have "0" at the l -th position.
- In the code array, each column contains at least one nonzero entry. Consider the l -th column of the code array.
- Let S_0 be the codewords with a "0" at the l -th position and S_1 be the codewords with a "1" at the l -th position.
- Let x be a codeword from S_1 . Adding x to each vector in S_0 , we obtain a set S'_1 of codewords with a "1" at the l -th position.

$$|S'_1| = |S_0| \text{ and } S'_1 \subseteq S_1$$

What we will get is a set containing 1 at l th location. Why, because S_0 is a set that has 0 at the l th location and x has, x is taken from the set S_1 so x has 1 at l th location. So if we add x to S_0 , elements in S_0 what we will get is there will be a 1 at the l th bit location. So we denote this class

(Refer Slide Time 11:32)

Linear block code

- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix \mathbf{G} contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.
- **Solution:** To prove that each column of this array has 2^{k-1} zeros and 2^{k-1} ones, we will show that the number of codewords that "1" at the l -th position is same as number of codewords that have "0" at the l -th position.
- In the code array, each column contains at least one nonzero entry. Consider the l -th column of the code array.
- Let S_0 be the codewords with a "0" at the l -th position and S_1 be the codewords with a "1" at the l -th position.
- Let \mathbf{x} be a codeword from S_1 . Adding \mathbf{x} to each vector in S_0 , we obtain a set S'_1 of codewords with a "1" at the l -th position.

$$|S'_1| = |S_0| \quad \text{and} \quad S'_1 \subseteq S_1$$

of codeword by S_1 prime and this S_1 prime will have 1 at the

(Refer Slide Time 11:40)

Linear block code

- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix \mathbf{G} contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.
- **Solution:** To prove that each column of this array has 2^{k-1} zeros and 2^{k-1} ones, we will show that the number of codewords that "1" at the l -th position is same as number of codewords that have "0" at the l -th position.
- In the code array, each column contains at least one nonzero entry. Consider the l -th column of the code array.
- Let S_0 be the codewords with a "0" at the l -th position and S_1 be the codewords with a "1" at the l -th position.
- Let \mathbf{x} be a codeword from S_1 . Adding \mathbf{x} to each vector in S_0 , we obtain a set S'_1 of codewords with a "1" at the l -th position.

$$|S'_1| = |S_0| \quad \text{and} \quad S'_1 \subseteq S_1$$

l th location and since this S_1 prime is generated by adding \mathbf{x} to this set of vectors in S_0 so number of elements in S_0 is going to be

(Refer Slide Time 11:52)

Linear block code

- **Problem 2 (contd.):** Consider an (n, k) linear code C whose generator matrix G contains no zero column. Arrange all the codewords of C as rows of a 2^k by n array
- b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.
- **Solution:** To prove that each column of this array has 2^{k-1} zeros and 2^{k-1} ones, we will show that the number of codewords that "1" at the l -th position is same as number of codewords that have "0" at the l -th position.
- In the code array, each column contains at least one nonzero entry. Consider the l -th column of the code array.
- Let S_0 be the codewords with a "0" at the l -th position and S_1 be the codewords with a "1" at the l -th position.
- Let x be a codeword from S_1 . Adding x to each vector in S_0 , we obtain a set S'_1 of codewords with a "1" at the l -th position.

$$|S'_1| = |S_0| \text{ and } S'_1 \subseteq S_1$$

same as number of elements in S_1 and S_1 is a subset of S_1 which is a set of all codewords which have 1 at l th location.

(Refer Slide Time 12:14)

Linear block code

- **Problem 2 (contd.):** The above condition implies that

$$|S_0| \leq |S_1| \quad (1)$$

So from this we get this condition that set of codewords which has 0 at l th location is less than or equal to set of

(Refer Slide Time 12:28)

Linear block code

- **Problem 2 (contd.):** The above condition implies that
$$|S_0| \leq |S_1| \quad (1)$$
- Adding x to each vector in S_1 , we obtain a set S'_0 of codewords with a "0" at the l -th position.
$$|S'_0| = |S_1| \quad \text{and} \quad S'_0 \subseteq S_0$$

codewords which has 1 at l th location. Now add the same vector x which has 1 at l th location to all the elements in S_1 . When we do that what we get is a new set of vectors which has 0 at l th location. We denote this set by S_0 prime. So S_0 prime is a set of codewords which are obtained by adding x to the set of vectors set of odd vectors which have 1 at l th location. So then we can write as set of vectors in S_0 prime is same as set of vectors in S_1 and since S_0 prime is a subset of S_0 what we can write then is,

(Refer Slide Time 13:26)

Linear block code

- **Problem 2 (contd.):** The above condition implies that
$$|S_0| \leq |S_1| \quad (1)$$
- Adding x to each vector in S_1 , we obtain a set S'_0 of codewords with a "0" at the l -th position.
$$|S'_0| = |S_1| \quad \text{and} \quad S'_0 \subseteq S_0$$
- The above condition implies that
$$|S_1| \leq |S_0| \quad (2)$$

from this relation and this relation we

(Refer Slide Time 13:30)

Linear block code

- **Problem 2 (contd.):** The above condition implies that
$$|S_0| \leq |S_1| \quad (1)$$
- Adding x to each vector in S_1 , we obtain a set S'_0 of codewords with a "0" at the l -th position.
$$|S'_0| = |S_1| \quad \text{and} \quad S'_0 \subseteq S_0$$
- The above condition implies that
$$|S_1| \leq |S_0| \quad (2)$$

can write

(Refer Slide Time 13:32)

Linear block code

- **Problem 2 (contd.):** The above condition implies that
$$|S_0| \leq |S_1| \quad (1)$$
- Adding x to each vector in S_1 , we obtain a set S'_0 of codewords with a "0" at the l -th position.
$$|S'_0| = |S_1| \quad \text{and} \quad S'_0 \subseteq S_0$$
- The above condition implies that
$$|S_1| \leq |S_0| \quad (2)$$

set of codewords which have 1 at l th location is less than set of codewords which has 0 at l th location. Now equation 1 and 2, they are going to be simultaneously

(Refer Slide Time 13:49)

Linear block code

- **Problem 2 (contd.):** The above condition implies that
$$|S_0| \leq |S_1| \quad (1)$$
- Adding x to each vector in S_1 , we obtain a set S'_0 of codewords with a "0" at the l -th position.
$$|S'_0| = |S_1| \quad \text{and} \quad S'_0 \subseteq S_0$$
- The above condition implies that
$$|S_1| \leq |S_0| \quad (2)$$

satisfied only when this is satisfied with equality. Then what it shows here is

(Refer Slide Time 13:57)

Linear block code

- **Problem 2 (contd.):** The above condition implies that
$$|S_0| \leq |S_1| \quad (1)$$
- Adding x to each vector in S_1 , we obtain a set S'_0 of codewords with a "0" at the l -th position.
$$|S'_0| = |S_1| \quad \text{and} \quad S'_0 \subseteq S_0$$
- The above condition implies that
$$|S_1| \leq |S_0| \quad (2)$$
- From (1) and (2), we get $|S_0| = |S_1|$. Therefore l -th column contains 2^{k-1} zeros and 2^{k-1} ones.

that at any l th location number of codewords which have 0 at l th location is same as number of codewords which have 1 at l th location. So basically each column will then have same number of 0s and

(Refer Slide Time 14:17)

Linear block code

- **Problem 2 (contd.):** The above condition implies that
$$|S_0| \leq |S_1| \quad (1)$$
- Adding x to each vector in S_1 , we obtain a set S'_0 of codewords with a "0" at the l -th position.
$$|S'_0| = |S_1| \quad \text{and} \quad S'_0 \subseteq S_0$$
- The above condition implies that
$$|S_1| \leq |S_0| \quad (2)$$
- From (1) and (2), we get $|S_0| = |S_1|$. Therefore l -th column contains 2^{k-1} zeros and 2^{k-1} ones.

same number of 1s.

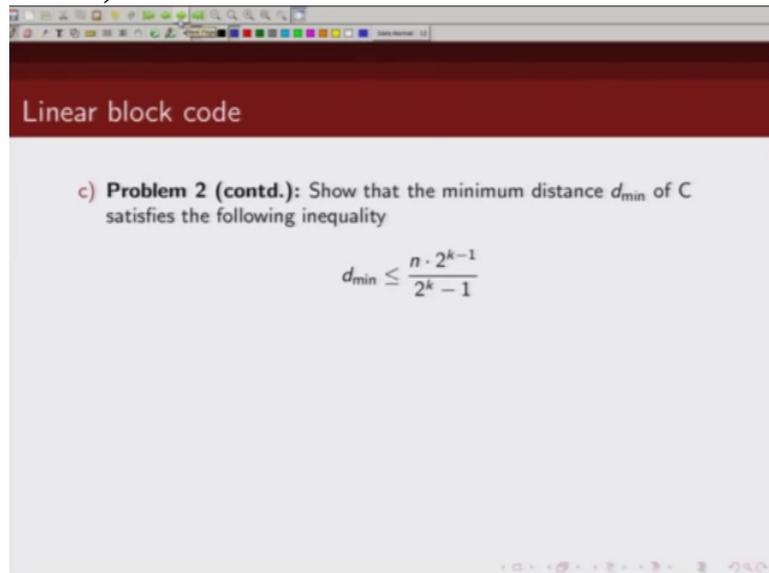
(Refer Slide Time 14:18)

Linear block code

- **Problem 2 (contd.):** The above condition implies that
$$|S_0| \leq |S_1| \quad (1)$$
- Adding x to each vector in S_1 , we obtain a set S'_0 of codewords with a "0" at the l -th position.
$$|S'_0| = |S_1| \quad \text{and} \quad S'_0 \subseteq S_0$$
- The above condition implies that
$$|S_1| \leq |S_0| \quad (2)$$
- From (1) and (2), we get $|S_0| = |S_1|$. Therefore l -th column contains 2^{k-1} zeros and 2^{k-1} ones.

Now we

(Refer Slide Time 14:20)



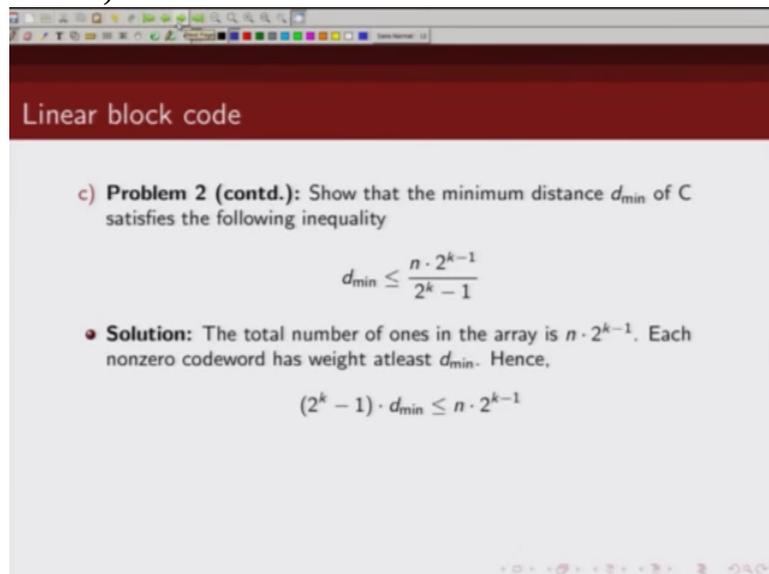
Linear block code

c) **Problem 2 (contd.):** Show that the minimum distance d_{\min} of C satisfies the following inequality

$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

prove another result. We show that minimum distance of the code is upper bounded by this quantity and to prove this result we are just going to use the result we just proved in the previous section. So

(Refer Slide Time 14:41)



Linear block code

c) **Problem 2 (contd.):** Show that the minimum distance d_{\min} of C satisfies the following inequality

$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

• **Solution:** The total number of ones in the array is $n \cdot 2^{k-1}$. Each nonzero codeword has weight atleast d_{\min} . Hence,

$$(2^k - 1) \cdot d_{\min} \leq n \cdot 2^{k-1}$$

in the previous section what we did was we arranged these 2^k codewords in an array two k cross n array and we showed that each column of this array has 2^k raised to power k minus 1 1s and 2^k raised to k minus 1 0s. So in this whole array which has n columns total number of 1s is given by this,

(Refer Slide Time 15:11)

Linear block code

c) **Problem 2 (contd.):** Show that the minimum distance d_{\min} of C satisfies the following inequality

$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

• **Solution:** The total number of ones in the array is $n \cdot 2^{k-1}$. Each nonzero codeword has weight atleast d_{\min} . Hence,

$$(2^k - 1) \cdot d_{\min} \leq n \cdot 2^{k-1}$$

n times 2 raised to power k minus 1 . Now since each non-zero codeword will have minimum distance at least d_{\min} and how many total codewords we have 2 raised to power k one of them is all zero codeword so how many non-zero codewords we have, that is given by 2^k minus 1

(Refer Slide Time 15:37)

Linear block code

c) **Problem 2 (contd.):** Show that the minimum distance d_{\min} of C satisfies the following inequality

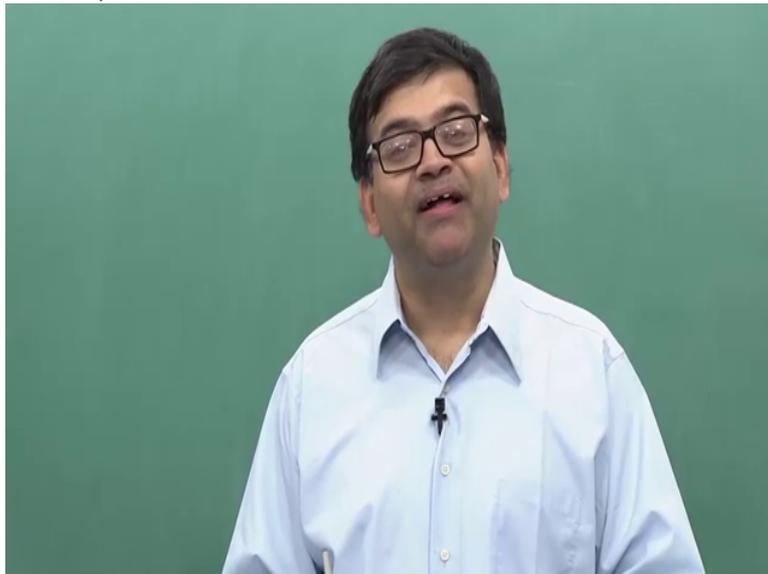
$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

• **Solution:** The total number of ones in the array is $n \cdot 2^{k-1}$. Each nonzero codeword has weight atleast d_{\min} . Hence,

$$\underline{(2^k - 1) \cdot d_{\min}} \leq n \cdot 2^{k-1}$$

and each of these non-zero codewords have minimum distance at least d_{\min} so total number of non-zero codewords multiplied by d_{\min} must be

(Refer Slide Time 15:53)



less than equal to total number of 1s

(Refer Slide Time 15:56)

Linear block code

c) **Problem 2 (contd.):** Show that the minimum distance d_{\min} of C satisfies the following inequality

$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$

• **Solution:** The total number of ones in the array is $n \cdot 2^{k-1}$. Each nonzero codeword has weight atleast d_{\min} . Hence,

$$(2^k - 1) \cdot d_{\min} \leq n \cdot 2^{k-1}$$

in this code array which is given by $n \cdot 2^{k-1}$, 2 raised to power k minus 1 . So from this relation then we can then write

(Refer Slide Time 16:09)

The slide is titled "Linear block code" in a dark red header. The main content is on a light gray background. It starts with a problem statement: "c) **Problem 2 (contd.):** Show that the minimum distance d_{\min} of C satisfies the following inequality". Below this is the inequality
$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$
. Then, a solution is provided: "• **Solution:** The total number of ones in the array is $n \cdot 2^{k-1}$. Each nonzero codeword has weight atleast d_{\min} . Hence," followed by the inequality
$$(2^k - 1) \cdot d_{\min} \leq n \cdot 2^{k-1}$$
. Finally, it states "• This implies that" followed by the same inequality
$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}$$
. The slide has a standard presentation navigation bar at the bottom.

that minimum distance of a code is upper bounded by this relationship.

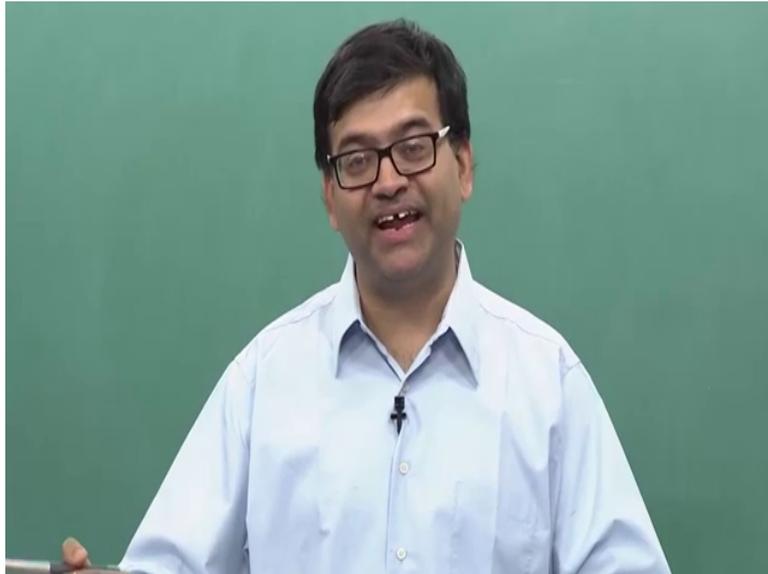
So

(Refer Slide Time 16:16)

The slide is titled "Minimum distance of a code" in a dark red header. The main content is on a light gray background. It contains a single bullet point: "• **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result." The slide has a standard presentation navigation bar at the bottom.

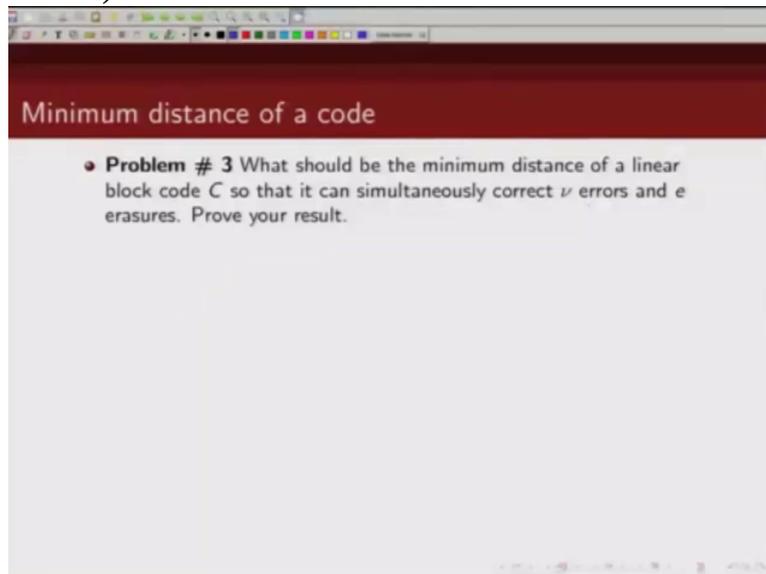
next problem that we will look at is what is a minimum distance of linear block C that can simultaneously correct μ errors and e erasures? Now just recall what do we mean by error correction and error

(Refer Slide Time 16:36)



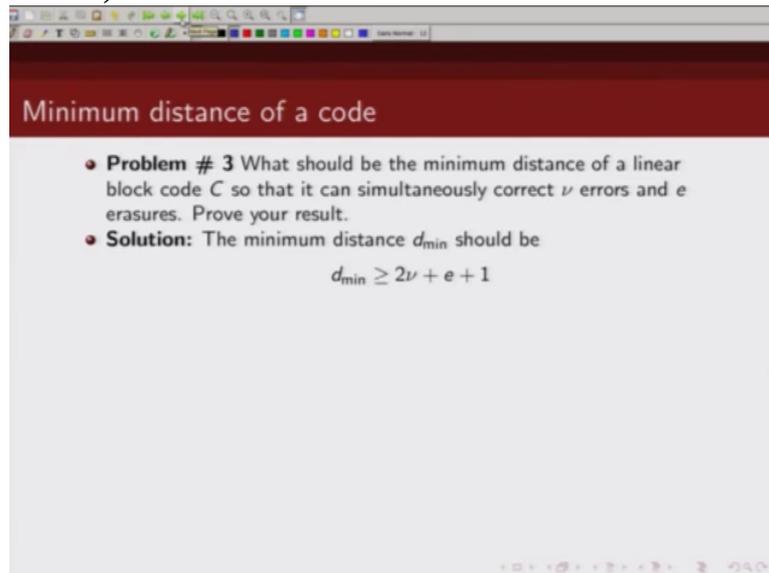
erasure correction. Basically, so erasure is basically, some of the bits are getting erased. So you send n bits. If e bits are getting erased what you are receiving is n minus e bits. And error correction you are familiar with, basically you want to correct errors that have happened in so many bit locations. So the question is

(Refer Slide Time 17:02)



what should be the minimum distance of a linear block code that can simultaneously correct μ errors as well as e erasures?

(Refer Slide Time 17:15)



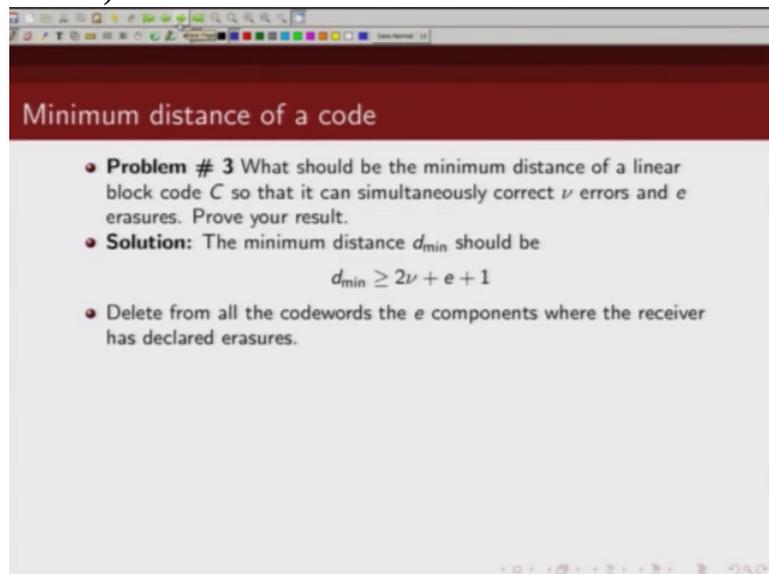
Minimum distance of a code

- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.
- **Solution:** The minimum distance d_{\min} should be
$$d_{\min} \geq 2\nu + e + 1$$

Now if the minimum distance of the code is at least two mu plus e plus 1 then it can simultaneously correct mu errors and e erasures. We are going to next prove this result.

So

(Refer Slide Time 17:31)

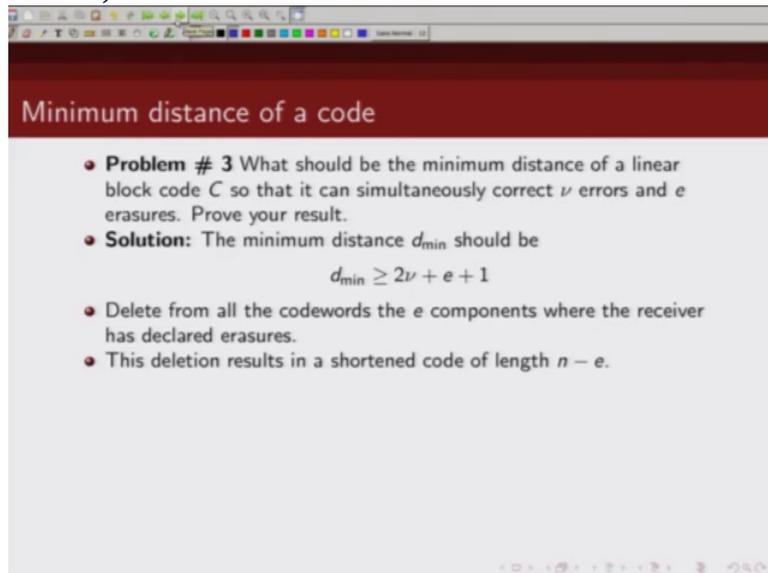


Minimum distance of a code

- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.
- **Solution:** The minimum distance d_{\min} should be
$$d_{\min} \geq 2\nu + e + 1$$
- Delete from all the codewords the e components where the receiver has declared erasures.

delete from all codewords e components which got erased. If we delete these e components what we are left is n minus e

(Refer Slide Time 17:46)

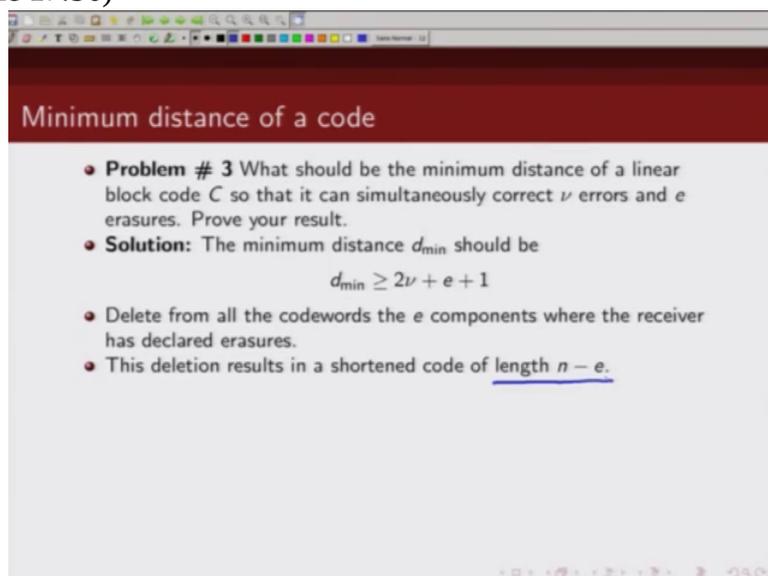


Minimum distance of a code

- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.
- **Solution:** The minimum distance d_{\min} should be
$$d_{\min} \geq 2\nu + e + 1$$
- Delete from all the codewords the e components where the receiver has declared erasures.
- This deletion results in a shortened code of length $n - e$.

length shortened codeword. So this deletion of e component results in a shortened code of length n minus e .

(Refer Slide Time 17:56)



Minimum distance of a code

- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.
- **Solution:** The minimum distance d_{\min} should be
$$d_{\min} \geq 2\nu + e + 1$$
- Delete from all the codewords the e components where the receiver has declared erasures.
- This deletion results in a shortened code of length $n - e$.

Now we

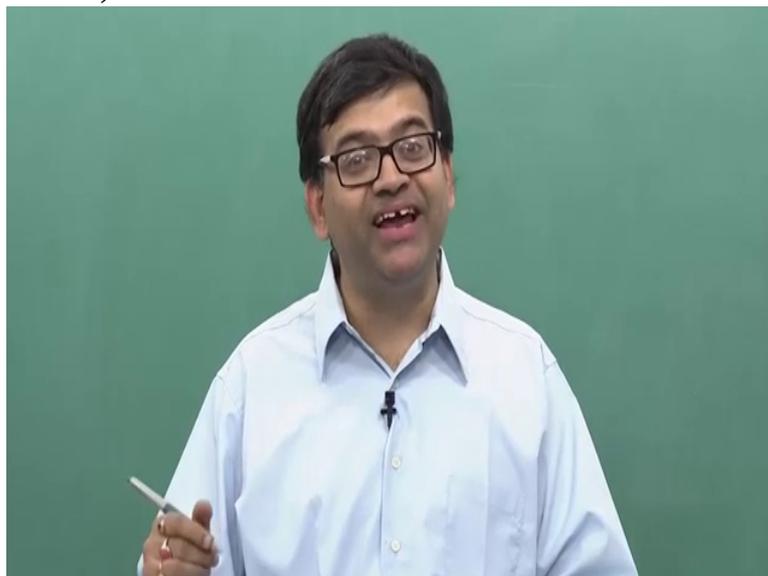
(Refer Slide Time 17:59)

Minimum distance of a code

- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.
- **Solution:** The minimum distance d_{\min} should be
$$d_{\min} \geq 2\nu + e + 1$$
- Delete from all the codewords the e components where the receiver has declared erasures.
- This deletion results in a shortened code of length $n - e$.
- The minimum distance of this shortened code should be atleast $d_{\min} - e \geq 2\nu + 1$.

know that if we want to correct t errors, what should be minimum distance of the code? It should be at least

(Refer Slide Time 18:06)



$2t + 1$. So this code

(Refer Slide Time 18:11)

Minimum distance of a code

- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.
- **Solution:** The minimum distance d_{\min} should be
$$d_{\min} \geq 2\nu + e + 1$$
- Delete from all the codewords the e components where the receiver has declared erasures.
- This deletion results in a shortened code of length $n - e$.
- The minimum distance of this shortened code should be atleast $d_{\min} - e \geq 2\nu + 1$.

basically, if we want to correct mu errors the minimum distance of the code after these e erasures

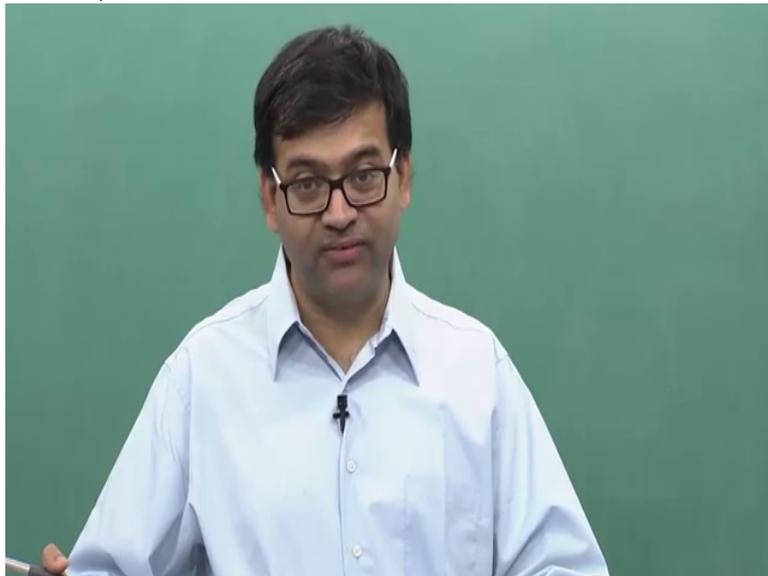
(Refer Slide Time 18:48)

Minimum distance of a code

- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.
- **Solution:** The minimum distance d_{\min} should be
$$d_{\min} \geq 2\nu + e + 1$$
- Delete from all the codewords the e components where the receiver has declared erasures.
- This deletion results in a shortened code of length $n - e$.
- The minimum distance of this shortened code should be atleast $d_{\min} - e \geq 2\nu + 1$.
- Hence, the ν errors in the unerased positions can be corrected. As a result the shortened code with e components erased can be recovered.

should be greater than equal to two mu plus 1. So if minimum distance of the code after these e erasures, the minimum distance is still larger than two mu plus 1, then this code can correct mu errors. So we want our minimum distance of the code to be at least 2 mu plus e plus 1. Now since these mu errors are in, in the unerased positions can be corrected if this condition holds so as a result basically we would be able to correct mu errors.

(Refer Slide Time 19:08)



Now remember we have to simultaneously

(Refer Slide Time 19:11)

Minimum distance of a code

- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.
- **Solution:** The minimum distance d_{\min} should be
$$d_{\min} \geq 2\nu + e + 1$$
- Delete from all the codewords the e components where the receiver has declared erasures.
- This deletion results in a shortened code of length $n - e$.
- The minimum distance of this shortened code should be atleast $d_{\min} - e \geq 2\nu + 1$.
- Hence, the ν errors in the unerased positions can be corrected. As a result the shortened code with e components erased can be recovered.

also be

(Refer Slide Time 19:14)

Minimum distance of a code

- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.
- **Solution:** The minimum distance d_{\min} should be
$$d_{\min} \geq 2\nu + e + 1$$
- Delete from all the codewords the e components where the receiver has declared erasures.
- This deletion results in a shortened code of length $n - e$.
- The minimum distance of this shortened code should be atleast $d_{\min} - e \geq 2\nu + 1$.
- Hence, the ν errors in the unerased positions can be corrected. As a result the shortened code with e components erased can be recovered.

able to basically, it would be not only simultaneously correct mu errors but we have to correct e erasures also. Now what is the condition on minimum distance such that e erasures can also be corrected? The minimum distance of the code should be at least greater than number of erasures plus 1. So if the minimum distance of the code

(Refer Slide Time 19:40)

Minimum distance of a code

- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.
- **Solution:** The minimum distance d_{\min} should be
$$d_{\min} \geq 2\nu + e + 1$$
- Delete from all the codewords the e components where the receiver has declared erasures.
- This deletion results in a shortened code of length $n - e$.
- The minimum distance of this shortened code should be atleast $d_{\min} - e \geq 2\nu + 1$.
- Hence, the ν errors in the unerased positions can be corrected. As a result the shortened code with e components erased can be recovered.
- Finally, since $d_{\min} \geq e + 1$, there is only one and only one codeword in the original code that agrees with the unerased components. Hence, the entire codeword can be recovered.

is greater than e plus 1

(Refer Slide Time 19:43)

Minimum distance of a code

- **Problem # 3** What should be the minimum distance of a linear block code C so that it can simultaneously correct ν errors and e erasures. Prove your result.
- **Solution:** The minimum distance d_{\min} should be
$$d_{\min} \geq 2\nu + e + 1$$
- Delete from all the codewords the e components where the receiver has declared erasures.
- This deletion results in a shortened code of length $n - e$.
- The minimum distance of this shortened code should be atleast $d_{\min} - e \geq 2\nu + 1$.
- Hence, the ν errors in the unerased positions can be corrected. As a result the shortened code with e components erased can be recovered.
- Finally, since $d_{\min} \geq e + 1$, there is only one and only one codeword in the original code that agrees with the unerased components. Hence, the entire codeword can be recovered.

then there is only one codeword in the original code that maps to the shortened code. So as long as minimum distance of the code is greater than $e + 1$, there is only one and one codeword in the original code that agrees with the unerased component. So there is as long as minimum distance of the code is greater than $e + 1$, there is only one code that maps from erased shortened code to the original code. And since in this case the d_{\min} is already $2\nu + e + 1$ which is greater than $e + 1$, this code would be able to correct e erasures as well. So if we choose our minimum distance of the code to be greater than equal to $2\nu + e + 1$, it would be able to correct ν errors as well as e erasures.

(Refer Slide Time 20:53)

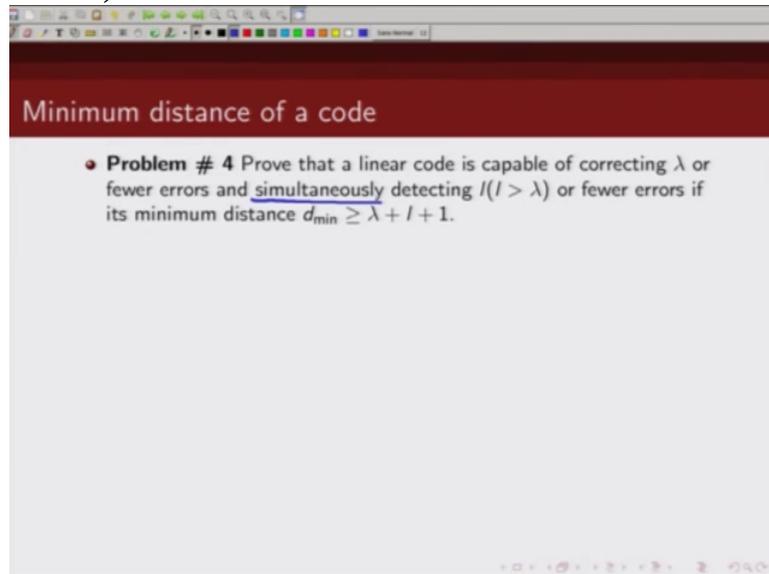
Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.

The next problem that we are going to solve is as follows. Prove that a linear block code is capable of correcting λ or fewer errors and simultaneously detecting l where l is greater

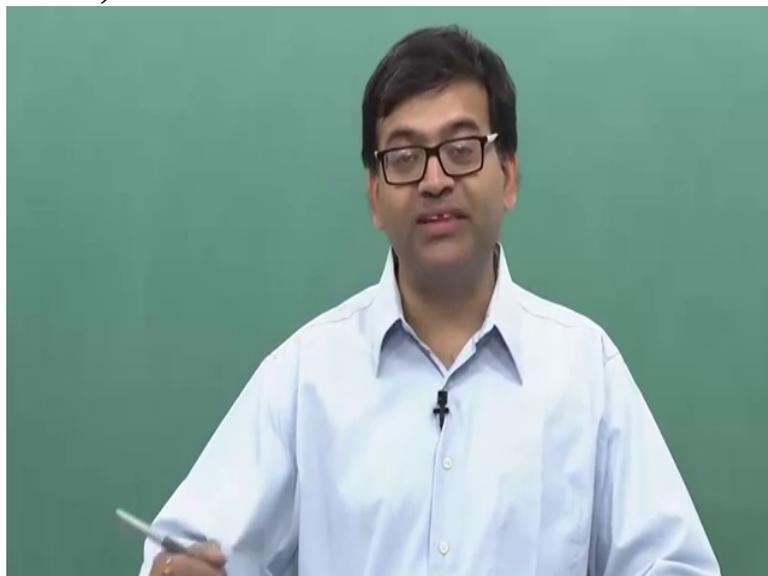
than λ or fewer errors if the minimum distance of the code is at least $\lambda + l + 1$. Please pay attention to the word simultaneously.

(Refer Slide Time 21:28)



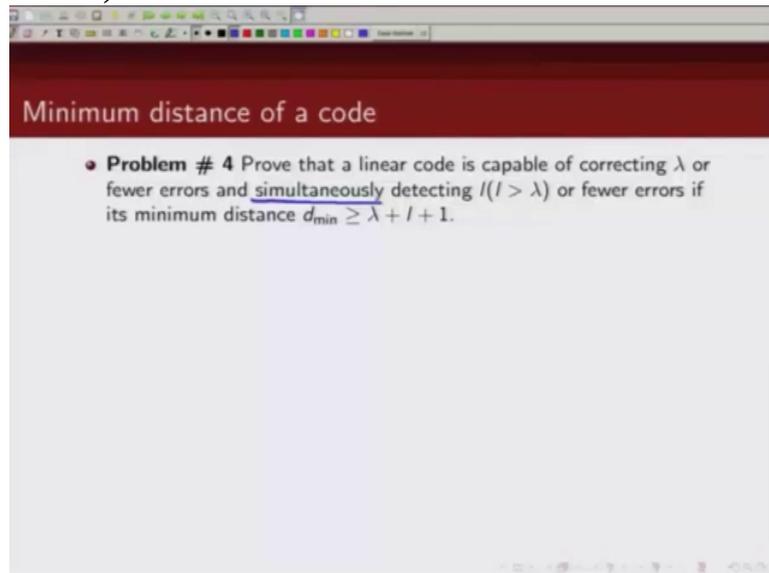
So we want not only to correct μ errors, along with that we should be able

(Refer Slide Time 21:34)



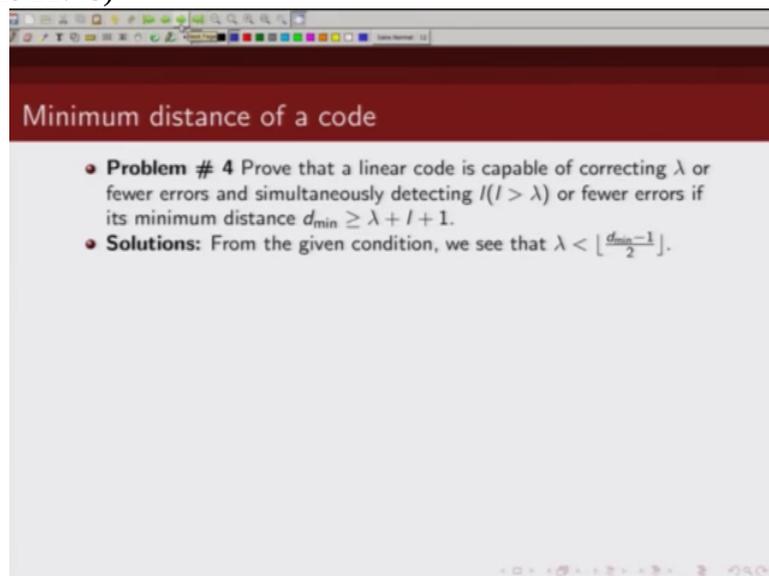
to detect l errors as well. That's what we mean by

(Refer Slide Time 21:40)



simultaneous error detection and correction. So let's prove this result. Now note

(Refer Slide Time 21:48)



lambda is less than l. So if minimum distance is lambda plus l plus 1, this is basically greater than 2 lambda plus 1.

(Refer Slide Time 22:05)

Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$. $\Rightarrow 2\lambda + l$
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min} - 1}{2} \rfloor$.

And if the minimum distance is greater than 2 lambda plus 1, it would be able to correct lambda errors. So from this given condition that d_{\min} is at least lambda plus l plus 1 where l is greater than lambda we know

(Refer Slide Time 22:25)

Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$. $\Rightarrow 2\lambda + l$
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min} - 1}{2} \rfloor$.

that the minimum distance is greater than 2 lambda plus 1.

(Refer Slide Time 22:30)



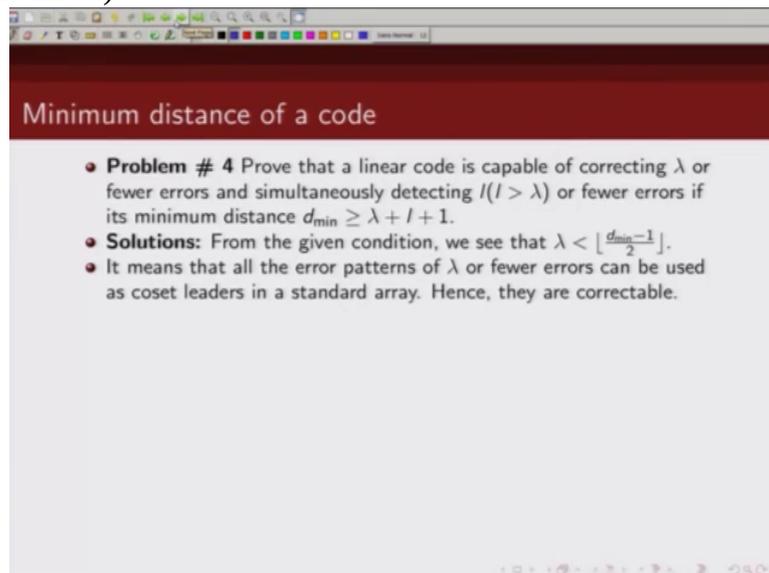
So it should be able to correct lambda errors.

(Refer Slide Time 22:34)

A screenshot of a presentation slide. The title is "Minimum distance of a code" in white text on a dark red background. Below the title, there are two bullet points. The first bullet point is "Problem # 4 Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l(l > \lambda)$ or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$." The second bullet point is "Solutions: From the given condition, we see that $\lambda < \lfloor \frac{d_{\min} - 1}{2} \rfloor$." The text " $\lambda < \lfloor \frac{d_{\min} - 1}{2} \rfloor$ " is underlined in blue. There is also a handwritten blue " $\geq 2\lambda + 1$ " next to the first bullet point.

Now note we want to,

(Refer Slide Time 22:36)



Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min} - 1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.

in addition to correcting lambda or fewer errors, we also want to simultaneously detect l errors. Now if we want to simultaneously detect those l or fewer errors we have to ensure that

(Refer Slide Time 22:53)



those error patterns of weight l or less are not in the same coset as the error patterns that we are

(Refer Slide Time 23:02)

Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.

trying to correct. Now since lambda errors can be corrected we can put all error patterns of lambda or fewer errors as coset leader in our standard array and they can be correctable.

(Refer Slide Time 23:22)

Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.

Next, to simultaneously detect l errors we have to show that none of these error patterns of weight l or less

(Refer Slide Time 23:32)



are in the same coset as these error

(Refer Slide Time 23:38)

A screenshot of a presentation slide. The title is "Minimum distance of a code" in white text on a dark red background. Below the title, there are three bullet points in black text on a white background. The first bullet point is "Problem # 4 Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$." The second bullet point is "Solutions: From the given condition, we see that $\lambda < \lfloor \frac{d_{\min} - 1}{2} \rfloor$." The third bullet point is "It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable." The fourth bullet point is "In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors."

patterns of lambda or less error. So we need to show that no error pattern x of length l or fewer are in the same coset as error pattern y of lambda or fewer errors. If they are in the same coset, because we are using those coset leaders for error correction, we would not be able to detect those error patterns. So it is important that those error patterns of weight l or less,

(Refer Slide Time 24:09)



if we want to detect them, they should not be in the same coset as the correctable error patterns.

(Refer Slide Time 24:17)

Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min} - 1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.

So

(Refer Slide Time 24:19)

Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies
$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

So we are now going to use method of contradiction to show that it is not possible to have these error pattern x of

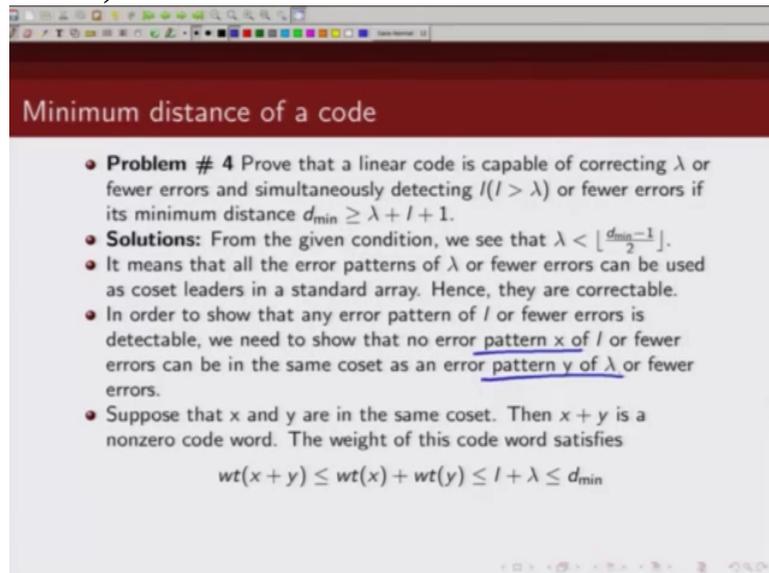
(Refer Slide Time 24:29)

Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies
$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

l or fewer errors in the same coset as these error patterns y

(Refer Slide Time 24:35)



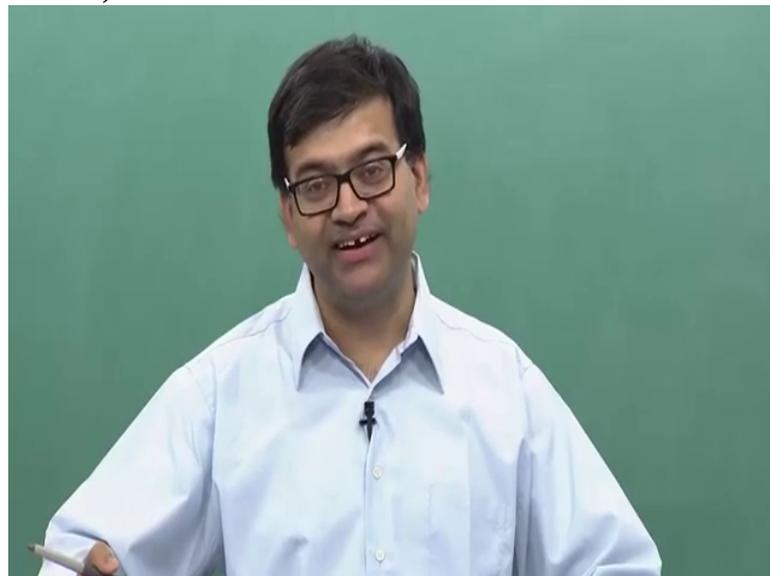
Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min} - 1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies

$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

of lambda or fewer errors which we are trying to correct. So how does this method of contradiction work? We will first assume that they are in the same coset and then we will show that this is not possible. Hence our assumption that they are in the same coset is wrong.

(Refer Slide Time 24:52)



(Refer Slide Time 24:54)

Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min} - 1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies
$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

So we start our proof by saying these error pattern x of weight l or less and error pattern y of weight λ or less, they are in the same coset. Now if x and y are in the same coset we know from our

(Refer Slide Time 25:16)



standard array that x plus y

(Refer Slide Time 25:19)

Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies

$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

should be a non-zero codeword. If you recall the entries of standard array we have in the first column an all zero codeword and then we had other codeword $v_2, v_3 \dots$ and then what we had was error pattern

(Refer Slide Time 25:37)

Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies $0, v_2, v_3, \dots$

$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

v_2 and then we had basically this was v_2 plus v_2 like that we had and if you add any two elements of a coset or a row

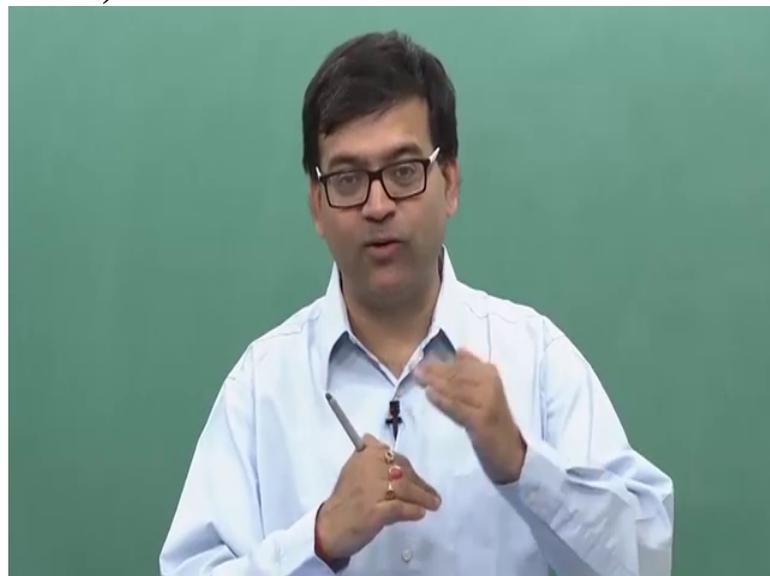
Refer Slide Time 25:52)

Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies $0 < wt(x+y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$ $\frac{0 \ v_2 \ v_3 \ \dots}{e_2 \ e_2 + v_2}$

what you will notice is sum of them is a valid codeword. So if x and y are in the same coset, $x + y$ must be a non-zero codeword. Now let's look at what is the weight of $x + y$. So weight of $x + y$ is less than equal to weight of x plus weight of y because it is possible that there are some common elements between x and y ,

(Refer Slide Time 26:25)



that's why the weight of $x + y$

(Refer Slide Time 26:27)

Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l(l > \lambda)$ or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies

$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

$\begin{matrix} 0 & v_1 & v_2 & \dots \\ e_1 & e_2 & e_2 + v_2 & \dots \end{matrix}$

is less than equal to weight of x plus weight of y . And what is weight of x ? x are the error pattern of weight l or less. So the maximum weight of x is l . Similarly maximum weight of y is λ . So weight of x plus y is then less than equal to λ plus l . And what is the minimum distance? Minimum distance code is at least λ plus l plus 1 . So weight of x plus y is then less than d_{\min} . So what we have shown is the weight of x plus y , x plus y should have been a codeword, is a codeword if they are in the same coset. If x and y are in the same coset, x plus y is a valid non-zero codeword but what we have shown here is weight of x plus y is less than d_{\min} so x plus y is a valid codeword. Its minimum weight

(Refer Slide Time 27:38)



should be at least d_{\min} . So

(Refer Slide Time 27:42)

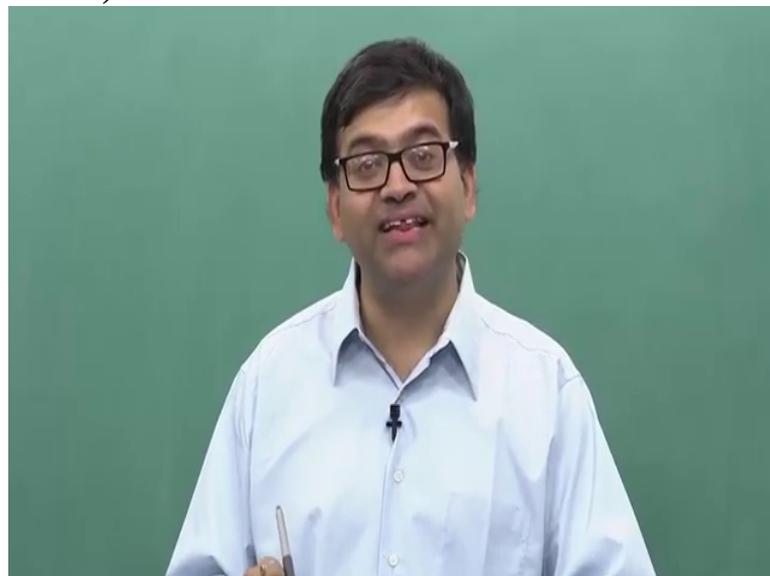
Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies $\frac{0 \ v_1 \ v_2 \ \dots}{e_1 \ e_2 + e_3}$

$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

from here, basically what we get is it is not possible to have x plus y in the same coset. Because if they were in the same coset x plus y would have been a valid codeword and its weight of x plus y should have been more than d_{\min} but here in this case it is coming out to be less than d_{\min} . Hence our assumption that x and y are in the same coset

(Refer Slide Time 28:13)



is wrong. Now if

(Refer Slide Time 28:17)

Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies

$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

*Handwritten note: $0 \ v_1 \ v_2 \ \dots$
 $e_2 \ e_2 + v_2$*

x and y are not in the same coset

(Refer Slide Time 28:20)

Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies

$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$

- This is impossible since the minimum weight of the code is d_{\min} . Hence x and y are in different cosets. As a result, when x occurs, it will not be mistaken as y . Therefore x is detectable.

then we can always put those error patterns of

(Refer Slide Time 28:26)



y and x in different cosets and hence we can simultaneously

(Refer Slide Time 28:33)

Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies
$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$
- This is impossible since the minimum weight of the code is d_{\min} . Hence x and y are in different cosets. As a result, when x occurs, it will not be mistaken as y . Therefore x is detectable.

detect and correct errors. So we can simultaneously correct lambda errors while detecting also l errors, Ok. So again to recap, basically we prove this result by showing that if you want to simultaneously correct and detect errors

(Refer Slide Time 28:56)



those error patterns should be in the different cosets and hence we can

(Refer Slide Time 29:05)

A screenshot of a presentation slide with a dark red header. The slide contains text and a mathematical equation. The text is as follows:

Minimum distance of a code

- **Problem # 4** Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.
- **Solutions:** From the given condition, we see that $\lambda < \lfloor \frac{d_{\min}-1}{2} \rfloor$.
- It means that all the error patterns of λ or fewer errors can be used as coset leaders in a standard array. Hence, they are correctable.
- In order to show that any error pattern of l or fewer errors is detectable, we need to show that no error pattern x of l or fewer errors can be in the same coset as an error pattern y of λ or fewer errors.
- Suppose that x and y are in the same coset. Then $x + y$ is a nonzero code word. The weight of this code word satisfies
$$wt(x + y) \leq wt(x) + wt(y) \leq l + \lambda \leq d_{\min}$$
- This is impossible since the minimum weight of the code is d_{\min} . Hence x and y are in different cosets. As a result, when x occurs, it will not be mistaken as y . Therefore x is detectable.

simultaneously correct and detect those error patterns.

(Refer Slide Time 29:10)

Minimum distance of a code

- **Problem # 5** Let C_i be the binary (n, k_i) linear code with generator matrix G_i and minimum distance d_i , respectively. Let C be the binary $(2n, k_1 + k_2)$ linear code with generator matrix

$$G = \begin{bmatrix} G_1 & G_1 \\ \mathbf{0} & G_2 \end{bmatrix}$$
 where $\mathbf{0}$ is a $k_2 \times n$ zero matrix. Calculate the minimum distance of C . Prove your result.

The next problem that we are going to solve is as follows. Let C_i be a binary linear code with code parameter given by n, k_i with generator matrix G_i and minimum distance d_i . And let us consider a new code C , a new binary code linear code of length $2n$ and message bit length $k_1 + k_2$ whose generator matrix is given by this expression. Now what is the minimum distance of this new code C ?

(Refer Slide Time 29:58)

Minimum distance of a code

- **Problem # 5** Let C_i be the binary (n, k_i) linear code with generator matrix G_i and minimum distance d_i , respectively. Let C be the binary $(2n, k_1 + k_2)$ linear code with generator matrix

$$G = \begin{bmatrix} G_1 & G_1 \\ \mathbf{0} & G_2 \end{bmatrix}$$
 where $\mathbf{0}$ is a $k_2 \times n$ zero matrix. Calculate the minimum distance of C . Prove your result.
- **Solution:** Let $\mathbf{u} = (u_0, u_1, \dots, u_{n-1})$ and $\mathbf{v} = (v_0, v_1, \dots, v_{n-1})$ be two binary n -tuples. We form $2n$ -tuple from \mathbf{u} and \mathbf{v} as follows

$$|\mathbf{u}| \mathbf{u} + \mathbf{v}| = (u_0, u_1, \dots, u_{n-1}, u_0 + v_0, u_1 + v_1, \dots, u_{n-1} + v_{n-1})$$

So to find out the minimum distance, so let's consider let \mathbf{u} and \mathbf{v} are two binary n -tuples. And we form a $2n$ -tuples as follows. If we look at this codeword \mathbf{v} how is this codeword generated? So it's one n -bit codeword, another n -bit codeword. First n -bit codeword is generated by using \mathbf{u} times \mathbf{v}_1 and second one you get basically \mathbf{u} times G_1 plus \mathbf{v} times G_2 .

So essentially the way you are generating this codeword, the first part contains n-bit codeword u and the second part is n-bit codeword which is

(Refer Slide Time 30:42)

Minimum distance of a code

- **Problem # 5** Let C_i be the binary (n, k_i) linear code with generator matrix G_i and minimum distance d_i , respectively. Let C be the binary $(2n, k_1 + k_2)$ linear code with generator matrix

$$G = \begin{bmatrix} G_1 & G_1 \\ \mathbf{0} & G_2 \end{bmatrix}$$
 where $\mathbf{0}$ is a $k_2 \times n$ zero matrix. Calculate the minimum distance of C . Prove your result.
- **Solution:** Let $\mathbf{u} = (u_0, u_1, \dots, u_{n-1})$ and $\mathbf{v} = (v_0, v_1, \dots, v_{n-1})$ be two binary n -tuples. We form $2n$ -tuple from \mathbf{u} and \mathbf{v} as follows

$$|\mathbf{u}|\mathbf{u} + \mathbf{v}| = (u_0, u_1, \dots, u_{n-1}, u_0 + v_0, u_1 + v_1, \dots, u_{n-1} + v_{n-1})$$

u plus v . So this $2n$ length codeword will be of the form like this

(Refer Slide Time 30:52)

Minimum distance of a code

- **Problem # 5** Let C_i be the binary (n, k_i) linear code with generator matrix G_i and minimum distance d_i , respectively. Let C be the binary $(2n, k_1 + k_2)$ linear code with generator matrix

$$G = \begin{bmatrix} G_1 & G_1 \\ \mathbf{0} & G_2 \end{bmatrix}$$
 where $\mathbf{0}$ is a $k_2 \times n$ zero matrix. Calculate the minimum distance of C . Prove your result.
- **Solution:** Let $\mathbf{u} = (u_0, u_1, \dots, u_{n-1})$ and $\mathbf{v} = (v_0, v_1, \dots, v_{n-1})$ be two binary n -tuples. We form $2n$ -tuple from \mathbf{u} and \mathbf{v} as follows

$$|\mathbf{u}|\mathbf{u} + \mathbf{v}| = (u_0, u_1, \dots, u_{n-1}, u_0 + v_0, u_1 + v_1, \dots, u_{n-1} + v_{n-1})$$

where the first n bits are $u_0, u_1, u_2, \dots, u_{n-1}$ and then next n bits are of the form $u_0 + v_0, u_1 + v_1$ and like that, so

(Refer Slide Time 31:07)

Minimum distance of a code

- **Problem # 5** Let C_1 be the binary (n, k_1) linear code with generator matrix G_1 and minimum distance d_1 , respectively. Let C_2 be the binary (n, k_2) linear code with generator matrix G_2 and minimum distance d_2 , respectively. Let C be the binary $(2n, k_1 + k_2)$ linear code with generator matrix

$$G = \begin{bmatrix} G_1 & G_2 \\ \mathbf{0} & G_1 \end{bmatrix}$$

where $\mathbf{0}$ is a $k_2 \times n$ zero matrix. Calculate the minimum distance of C . Prove your result.

- **Solution:** Let $\mathbf{u} = (u_0, u_1, \dots, u_{n-1})$ and $\mathbf{v} = (v_0, v_1, \dots, v_{n-1})$ be two binary n -tuples. We form $2n$ -tuple from \mathbf{u} and \mathbf{v} as follows

$$|\mathbf{u}|\mathbf{u} + \mathbf{v}| = (u_0, u_1, \dots, u_{n-1}, u_0 + v_0, u_1 + v_1, \dots, u_{n-1} + v_{n-1})$$

- The linear block code C is

$$\begin{aligned} C &= |C_1|C_1 + C_2| \\ &= \{|\mathbf{u}|\mathbf{u} + \mathbf{v}| : \mathbf{u} \in C_1, \text{ and } \mathbf{v} \in C_2\} \end{aligned}$$

as I said our linear block code, see the new code of length $2n$ can be written of this form where you have a code \mathbf{u} of length n which belongs to C_1 and then the second part which is n bit part is $\mathbf{u} + \mathbf{v}$ where \mathbf{u} belongs to C_1 and \mathbf{v} belongs to C_2 .

Now we will

(Refer Slide Time 31:33)

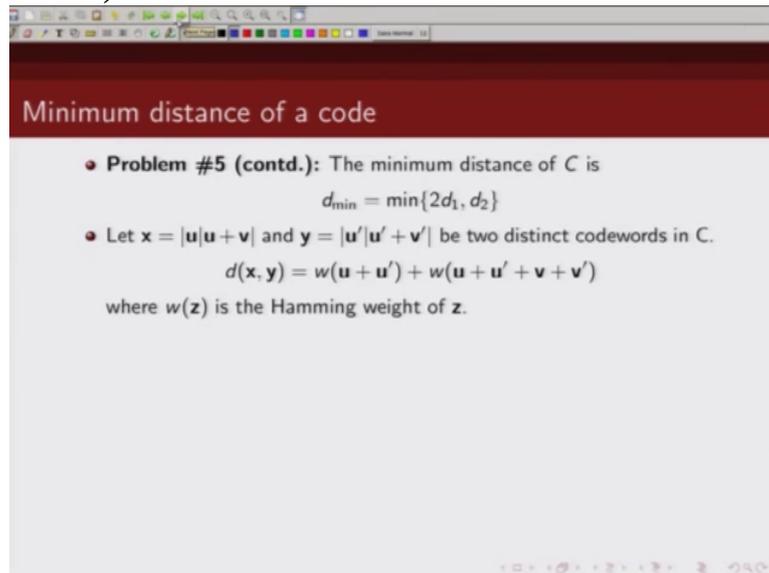
Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$

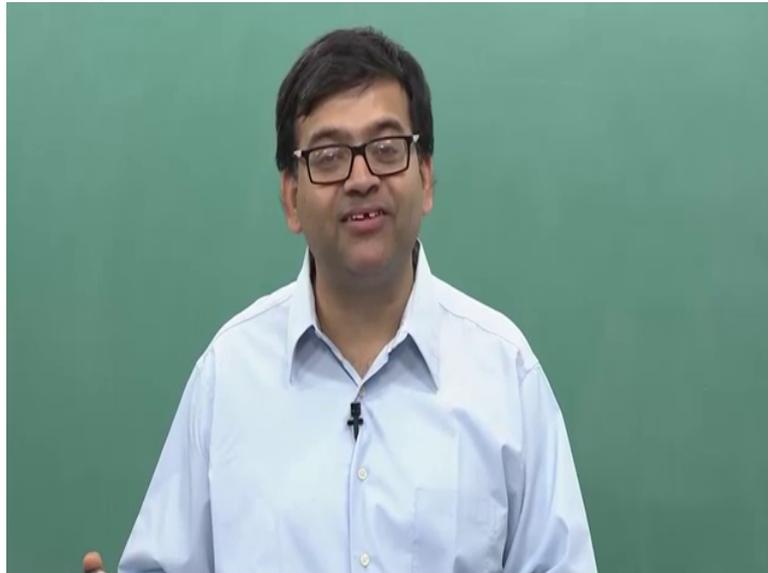
show that minimum distance of the code is minimum of two d_1 or d_2 where d_2 is the minimum distance of the code n, k_2 and d_1 is the minimum distance of the code n, k_1 . So

(Refer Slide Time 31:51)



let us consider two distinct codewords x and y . So x we denote as concatenation of u and u plus v and this is u prime, u prime plus v prime. Let x and y be two distinct codewords in C . Now what is the Hamming distance

(Refer Slide Time 32:16)



between x and y ? We can write down the Hamming distance between x and y as

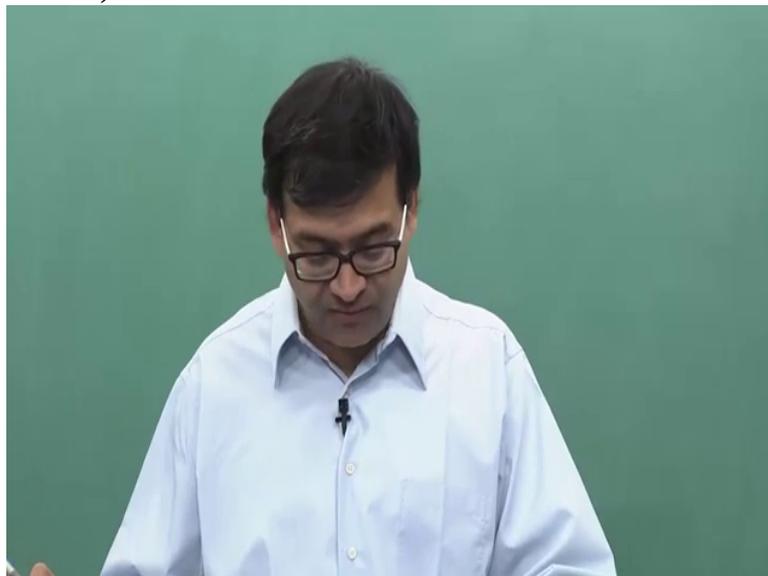
(Refer Slide Time 32:23)

Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is
$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $x = |u|u+v|$ and $y = |u'|u'+v'|$ be two distinct codewords in C .
$$d(x, y) = w(u+u') + w(u+u'+v+v')$$
where $w(z)$ is the Hamming weight of z .

the Hamming weight between u plus u prime plus Hamming weight between u plus v plus v hat plus u hat. So the Hamming distance between x and y can be written as Hamming weight of u plus u hat plus Hamming weight of u plus u hat plus v plus v hat.

(Refer Slide Time 32:50)



(Refer Slide Time 32:52)

Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is
$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = \mathbf{u}|\mathbf{u} + \mathbf{v}$ and $\mathbf{y} = \mathbf{u}'|\mathbf{u}' + \mathbf{v}'$ be two distinct codewords in C .
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}')$$
where $w(\mathbf{z})$ is the Hamming weight of \mathbf{z} .

Now note

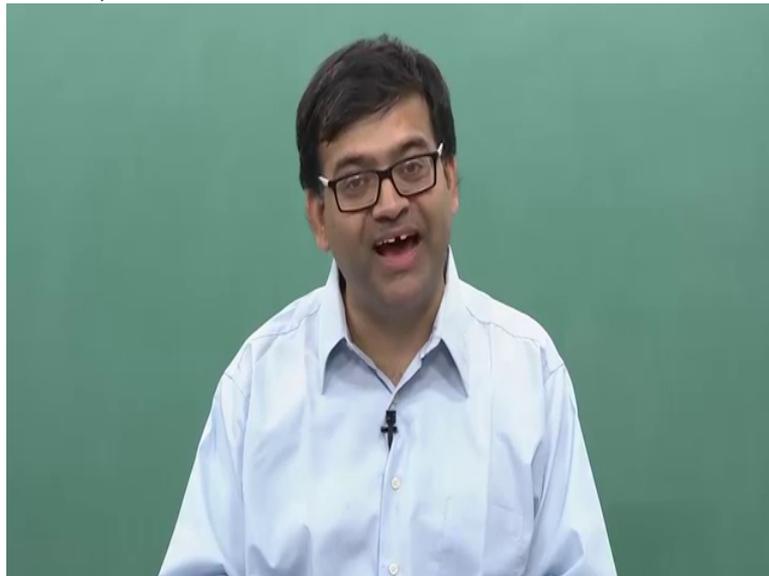
(Refer Slide Time 32:54)

Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is
$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = \mathbf{u}|\mathbf{u} + \mathbf{v}$ and $\mathbf{y} = \mathbf{u}'|\mathbf{u}' + \mathbf{v}'$ be two distinct codewords in C .
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}')$$
where $w(\mathbf{z})$ is the Hamming weight of \mathbf{z} .
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}')$$

\mathbf{x} and \mathbf{y} are distinct codewords. So let us consider two scenarios. In first case we will consider \mathbf{v} is same as \mathbf{v} prime. In second case we will consider \mathbf{v} is not same as \mathbf{v} prime. So if we consider \mathbf{v} as same as \mathbf{v} prime, since \mathbf{x} and \mathbf{y} are distinct codewords what we will have is \mathbf{u} is not same as

(Refer Slide Time 33:22)



u prime. So in this case the Hamming

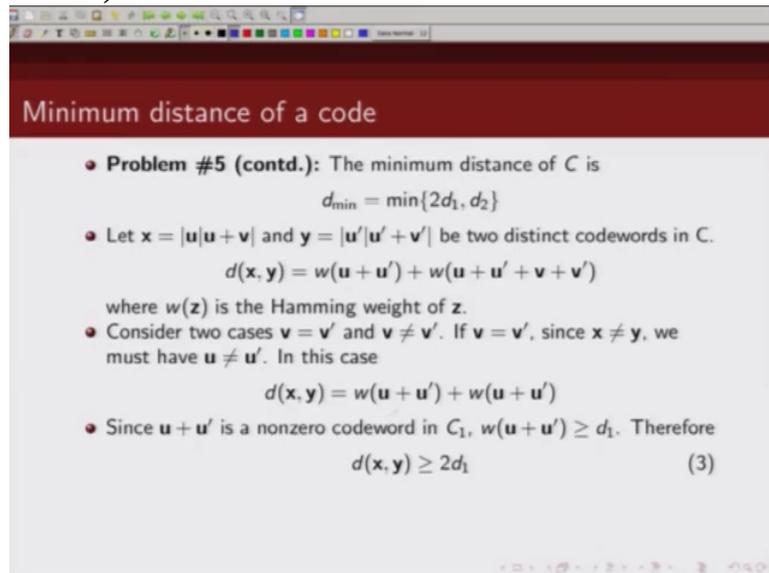
(Refer Slide Time 33:26)

Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is
$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = \mathbf{u}|\mathbf{u} + \mathbf{v}|$ and $\mathbf{y} = \mathbf{u}'|\mathbf{u}' + \mathbf{v}'|$ be two distinct codewords in C .
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}')$$
where $w(\mathbf{z})$ is the Hamming weight of \mathbf{z} .
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}')$$

distance between \mathbf{x} and \mathbf{y} will be given by Hamming weight of \mathbf{u} plus \mathbf{u} prime plus, now since \mathbf{v} and \mathbf{v} prime are same this will be zero so this will be same as Hamming weight of \mathbf{u} plus \mathbf{u} prime. So then in this case when \mathbf{v} is same as \mathbf{v} prime we can write the Hamming distance between \mathbf{x} and \mathbf{y} as Hamming weight of \mathbf{u} plus \mathbf{u} prime plus Hamming weight of \mathbf{u} plus \mathbf{u} prime. And since what is \mathbf{u} plus \mathbf{u} prime?

(Refer Slide Time 34:03)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is
$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = \mathbf{u}|\mathbf{u} + \mathbf{v}$ and $\mathbf{y} = \mathbf{u}'|\mathbf{u}' + \mathbf{v}'$ be two distinct codewords in C .
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}')$$
where $w(\mathbf{z})$ is the Hamming weight of \mathbf{z} .
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}')$$
- Since $\mathbf{u} + \mathbf{u}'$ is a nonzero codeword in C_1 , $w(\mathbf{u} + \mathbf{u}') \geq d_1$. Therefore
$$d(\mathbf{x}, \mathbf{y}) \geq 2d_1 \quad (3)$$

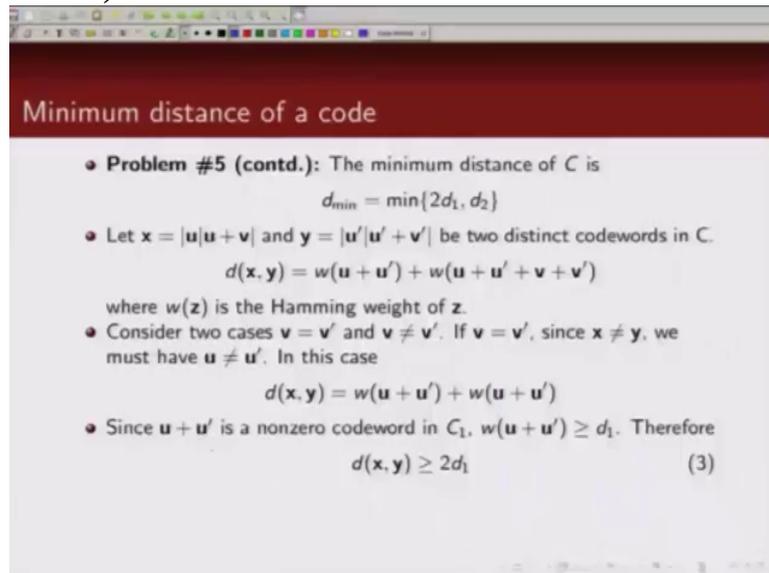
\mathbf{u} and \mathbf{u}' are two codewords belonging to

(Refer Slide Time 34:08)



C_1 . So sum of two codewords for a linear block code is another valid codeword. So $\mathbf{u} + \mathbf{u}'$ is going to be

(Refer Slide Time 34:18)

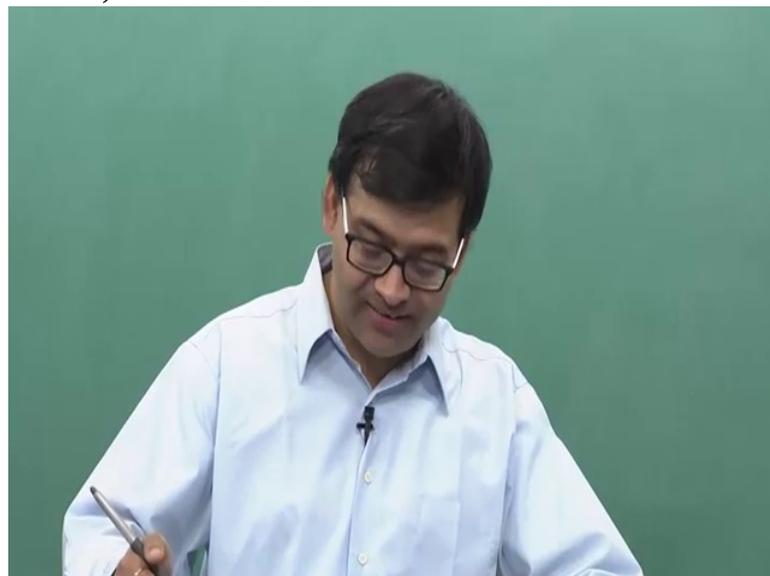


Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is
$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = \mathbf{u}|\mathbf{u} + \mathbf{v}|$ and $\mathbf{y} = \mathbf{u}'|\mathbf{u}' + \mathbf{v}'|$ be two distinct codewords in C .
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}')$$
where $w(\mathbf{z})$ is the Hamming weight of \mathbf{z} .
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}')$$
- Since $\mathbf{u} + \mathbf{u}'$ is a nonzero codeword in C_1 , $w(\mathbf{u} + \mathbf{u}') \geq d_1$. Therefore
$$d(\mathbf{x}, \mathbf{y}) \geq 2d_1 \quad (3)$$

another valid codeword. So then, so what would be the minimum distance of $\mathbf{u} + \mathbf{u}'$? It would be at least the minimum distance of the code C_1 which is d_1 .

(Refer Slide Time 34:31)



So then Hamming distance between \mathbf{x} and \mathbf{y} would be

(Refer Slide Time 34:36)

Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is
$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = \mathbf{u}|\mathbf{u} + \mathbf{v}|$ and $\mathbf{y} = \mathbf{u}'|\mathbf{u}' + \mathbf{v}'|$ be two distinct codewords in C .
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}')$$
where $w(\mathbf{z})$ is the Hamming weight of \mathbf{z} .
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}')$$
- Since $\mathbf{u} + \mathbf{u}'$ is a nonzero codeword in C_1 , $w(\mathbf{u} + \mathbf{u}') \geq d_1$. Therefore
$$d(\mathbf{x}, \mathbf{y}) \geq 2d_1 \quad (3)$$

greater than equal to 2 times d_1 . So for the case when v is equal to v prime we have shown that minimum distance

(Refer Slide Time 34:49)

Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is
$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = \mathbf{u}|\mathbf{u} + \mathbf{v}|$ and $\mathbf{y} = \mathbf{u}'|\mathbf{u}' + \mathbf{v}'|$ be two distinct codewords in C .
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}')$$
where $w(\mathbf{z})$ is the Hamming weight of \mathbf{z} .
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}')$$
- Since $\mathbf{u} + \mathbf{u}'$ is a nonzero codeword in C_1 , $w(\mathbf{u} + \mathbf{u}') \geq d_1$. Therefore
$$\underline{d(\mathbf{x}, \mathbf{y}) \geq 2d_1} \quad (3)$$

should be at least 2 times d_1 .

(Refer Slide Time 34:53)



Now let us consider the case when

(Refer Slide Time 34:58)

Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is
$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = \mathbf{u}|\mathbf{u} + \mathbf{v}|$ and $\mathbf{y} = \mathbf{u}'|\mathbf{u}' + \mathbf{v}'|$ be two distinct codewords in C .
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}')$$
where $w(\mathbf{z})$ is the Hamming weight of \mathbf{z} .
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}')$$
- Since $\mathbf{u} + \mathbf{u}'$ is a nonzero codeword in C_1 , $w(\mathbf{u} + \mathbf{u}') \geq d_1$. Therefore
$$\underline{d(\mathbf{x}, \mathbf{y})} \geq 2d_1 \quad (3)$$

\mathbf{v} is not equal to \mathbf{v}' . So before that we will just

(Refer Slide Time 35:05)

Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = \mathbf{u}|\mathbf{u} + \mathbf{v}|$ and $\mathbf{y} = \mathbf{u}'|\mathbf{u}' + \mathbf{v}'|$ be two distinct codewords in C .

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}')$$
 where $w(\mathbf{z})$ is the Hamming weight of \mathbf{z} .
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}')$$
- Since $\mathbf{u} + \mathbf{u}'$ is a nonzero codeword in C_1 , $w(\mathbf{u} + \mathbf{u}') \geq d_1$. Therefore

$$d(\mathbf{x}, \mathbf{y}) \geq 2d_1 \quad (3)$$
- From triangle inequality, we have

$$d(\mathbf{x}, \mathbf{y}) \geq d(\mathbf{x}, \mathbf{z}) - d(\mathbf{y}, \mathbf{z})$$

$$w(\mathbf{x} + \mathbf{y}) \geq wt(\mathbf{x} + \mathbf{z}) - wt(\mathbf{y} + \mathbf{z})$$

state again the triangular inequality that we are going to use; so from the triangular inequality we know that Hamming distance between x and y is greater than equal to Hamming distance between x and z minus Hamming distance between y and z . And we know the Hamming distance is nothing but Hamming weight of x plus y , Hamming weight of x plus y and Hamming weight of x plus z minus Hamming weight of

(Refer Slide Time 35:36)

Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = \mathbf{u}|\mathbf{u} + \mathbf{v}|$ and $\mathbf{y} = \mathbf{u}'|\mathbf{u}' + \mathbf{v}'|$ be two distinct codewords in C .

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}')$$
 where $w(\mathbf{z})$ is the Hamming weight of \mathbf{z} .
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}')$$
- Since $\mathbf{u} + \mathbf{u}'$ is a nonzero codeword in C_1 , $w(\mathbf{u} + \mathbf{u}') \geq d_1$. Therefore

$$d(\mathbf{x}, \mathbf{y}) \geq 2d_1 \quad (3)$$
- From triangle inequality, we have

$$d(\mathbf{x}, \mathbf{y}) \geq d(\mathbf{x}, \mathbf{z}) - d(\mathbf{y}, \mathbf{z})$$

$$w(\mathbf{x} + \mathbf{y}) \geq wt(\mathbf{x} + \mathbf{z}) - wt(\mathbf{y} + \mathbf{z})$$

y plus z . So we can write this expression in terms of Hamming distance or we can write in terms of Hamming weight. Now let us take x plus z to be

(Refer Slide Time 35:48)

Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get

$$w(u + u' + v + v') \geq w(v + v') - w(u + u')$$

equal to $v + v'$ and $y + z = u + u'$. And we put these values of x

(Refer Slide Time 35:58)

Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $x = |u|u + v|$ and $y = |u'|u' + v'|$ be two distinct codewords in C .

$$d(x, y) = w(u + u') + w(u + u' + v + v')$$
 where $w(z)$ is the Hamming weight of z .
- Consider two cases $v = v'$ and $v \neq v'$. If $v = v'$, since $x \neq y$, we must have $u \neq u'$. In this case

$$d(x, y) = w(u + u') + w(u + u')$$
- Since $u + u'$ is a nonzero codeword in C_1 , $w(u + u') \geq d_1$. Therefore

$$d(x, y) \geq 2d_1 \quad (3)$$
- From triangle inequality, we have

$$d(x, y) \geq d(x, z) - d(y, z)$$

$$w(x + y) \geq w(x + z) - w(y + z)$$

and $x + y$ and $x + z$ and $y + z$, we put these values

(Refer Slide Time 36:05)

Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get

$$w(u + u' + v + v') \geq w(v + v') - w(u + u')$$

in this expression.

(Refer Slide Time 36:07)

Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $x = |u|u + v|$ and $y = |u'|u' + v'|$ be two distinct codewords in C .

$$d(x, y) = w(u + u') + w(u + u' + v + v')$$
 where $w(z)$ is the Hamming weight of z .
- Consider two cases $v = v'$ and $v \neq v'$. If $v = v'$, since $x \neq y$, we must have $u \neq u'$. In this case

$$d(x, y) = w(u + u') + w(u + u')$$
- Since $u + u'$ is a nonzero codeword in C_1 , $w(u + u') \geq d_1$. Therefore

$$d(x, y) \geq 2d_1 \tag{3}$$
- From triangle inequality, we have

$$d(x, y) \geq d(x, z) - d(y, z)$$

$$w(x + y) \geq w(x + z) - w(y + z)$$

So what is x plus y? x plus y

(Refer Slide Time 36:12)

Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get

$$w(u + u' + v + v') \geq w(v + v') - w(u + u')$$

would be v plus v prime plus u plus u prime so x plus y is basically u plus u prime plus v plus v prime. So from here weight of

(Refer Slide Time 36:27)

Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $x = |u|u + v|$ and $y = |u'|u' + v'|$ be two distinct codewords in C .

$$d(x, y) = w(u + u') + w(u + u' + v + v')$$
 where $w(z)$ is the Hamming weight of z .
- Consider two cases $v = v'$ and $v \neq v'$. If $v = v'$, since $x \neq y$, we must have $u \neq u'$. In this case

$$d(x, y) = w(u + u') + w(u + u')$$
- Since $u + u'$ is a nonzero codeword in C_1 , $w(u + u') \geq d_1$. Therefore

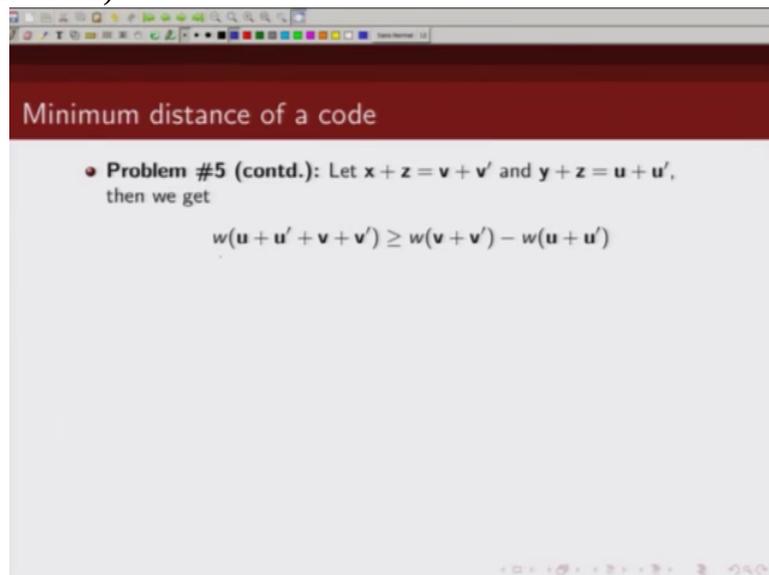
$$d(x, y) \geq 2d_1 \quad (3)$$
- From triangle inequality, we have

$$d(x, y) \geq d(x, z) - d(y, z)$$

$$w(x + y) \geq w(x + z) - w(y + z)$$

x plus y is given

(Refer Slide Time 36:30)



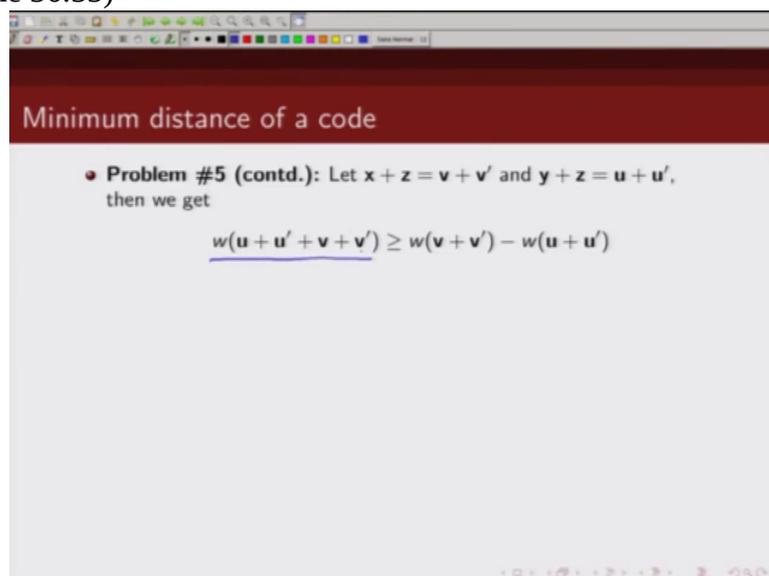
Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get

$$w(u + u' + v + v') \geq w(v + v') - w(u + u')$$

by this.

(Refer Slide Time 36:33)



Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get

$$\underline{w(u + u' + v + v')} \geq w(v + v') - w(u + u')$$

Next what we had was weight of

(Refer Slide Time 36:36)

Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u} + \mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}' + \mathbf{v}'|$ be two distinct codewords in C .

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}')$$
 where $w(\mathbf{z})$ is the Hamming weight of \mathbf{z} .
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}')$$
- Since $\mathbf{u} + \mathbf{u}'$ is a nonzero codeword in C_1 , $w(\mathbf{u} + \mathbf{u}') \geq d_1$. Therefore

$$d(\mathbf{x}, \mathbf{y}) \geq 2d_1 \quad (3)$$
- From triangle inequality, we have

$$d(\mathbf{x}, \mathbf{y}) \geq d(\mathbf{x}, \mathbf{z}) - d(\mathbf{y}, \mathbf{z})$$

$$w(\mathbf{x} + \mathbf{y}) \geq w(\mathbf{x} + \mathbf{z}) - w(\mathbf{y} + \mathbf{z})$$

\mathbf{x} plus \mathbf{z} . What is weight of \mathbf{x} plus \mathbf{z} ?

(Refer Slide Time 36:40)

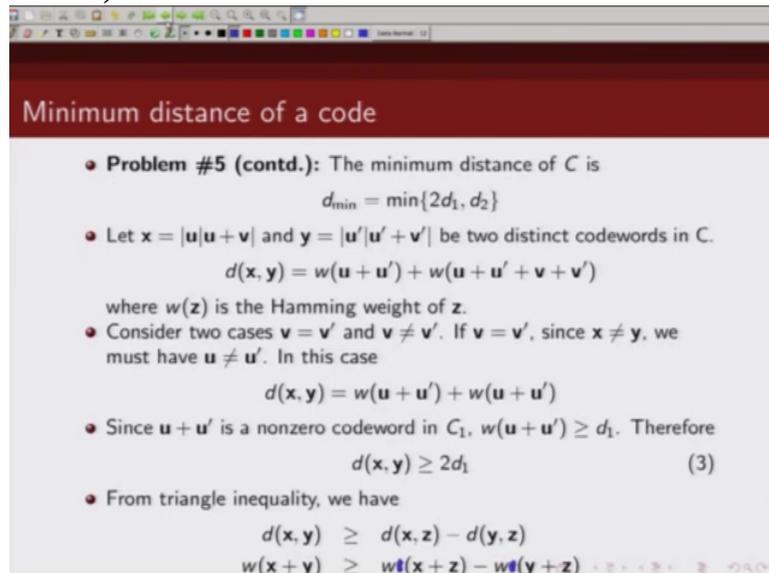
Minimum distance of a code

- **Problem #5 (contd.):** Let $\mathbf{x} + \mathbf{z} = \mathbf{v} + \mathbf{v}'$ and $\mathbf{y} + \mathbf{z} = \mathbf{u} + \mathbf{u}'$, then we get

$$w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}') \geq w(\mathbf{v} + \mathbf{v}') - w(\mathbf{u} + \mathbf{u}')$$

Weight of \mathbf{x} plus \mathbf{z} is weight of \mathbf{v} plus \mathbf{v} prime and similarly weight of \mathbf{y} plus \mathbf{z} is given by this, Ok. So this is upper bounded, this is lower bounded by this quantity. This is lower bounded by this quantity. Now go back and

(Refer Slide Time 37:06)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = \mathbf{u}|\mathbf{u} + \mathbf{v}|$ and $\mathbf{y} = \mathbf{u}'|\mathbf{u}' + \mathbf{v}'|$ be two distinct codewords in C .

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}')$$
 where $w(\mathbf{z})$ is the Hamming weight of \mathbf{z} .
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}')$$
- Since $\mathbf{u} + \mathbf{u}'$ is a nonzero codeword in C_1 , $w(\mathbf{u} + \mathbf{u}') \geq d_1$. Therefore

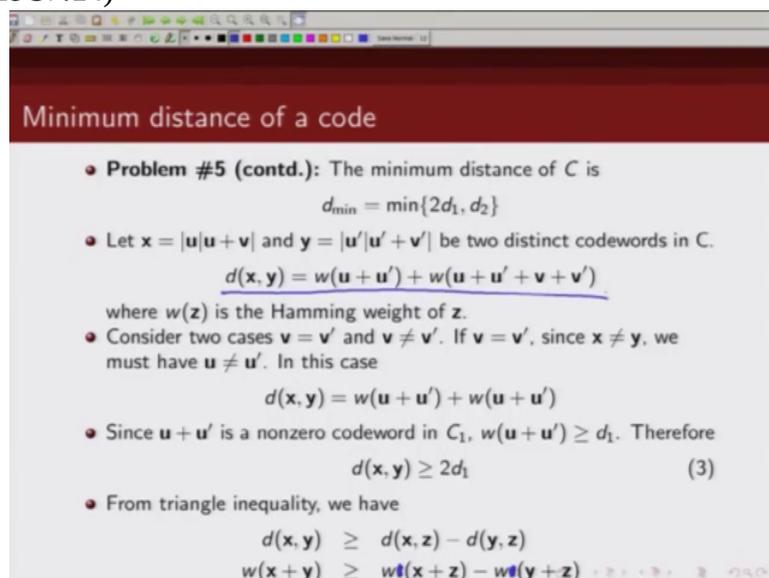
$$d(\mathbf{x}, \mathbf{y}) \geq 2d_1 \quad (3)$$
- From triangle inequality, we have

$$d(\mathbf{x}, \mathbf{y}) \geq d(\mathbf{x}, \mathbf{z}) - d(\mathbf{y}, \mathbf{z})$$

$$w(\mathbf{x} + \mathbf{y}) \geq w(\mathbf{x} + \mathbf{z}) - w(\mathbf{y} + \mathbf{z})$$

see what is our minimum distance between \mathbf{x} and \mathbf{y} ? Minimum distance between \mathbf{x} and \mathbf{y} is given by

(Refer Slide Time 37:14)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = \mathbf{u}|\mathbf{u} + \mathbf{v}|$ and $\mathbf{y} = \mathbf{u}'|\mathbf{u}' + \mathbf{v}'|$ be two distinct codewords in C .

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}')$$
 where $w(\mathbf{z})$ is the Hamming weight of \mathbf{z} .
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}')$$
- Since $\mathbf{u} + \mathbf{u}'$ is a nonzero codeword in C_1 , $w(\mathbf{u} + \mathbf{u}') \geq d_1$. Therefore

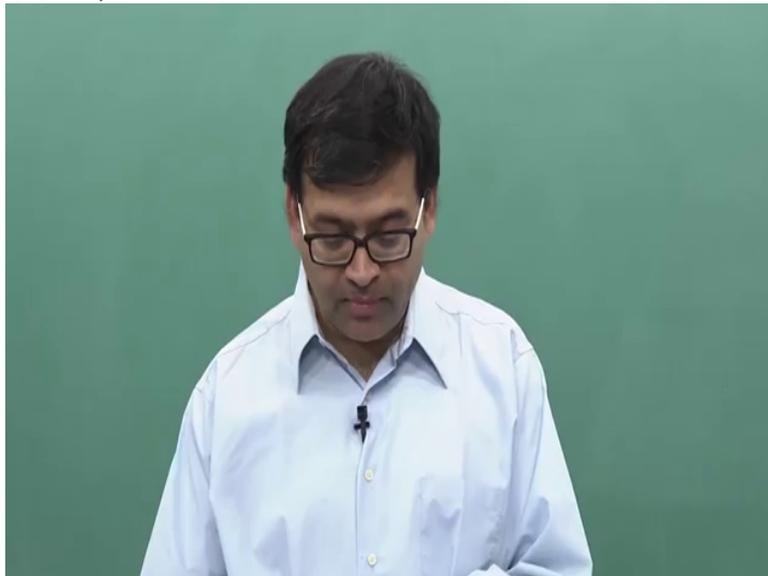
$$d(\mathbf{x}, \mathbf{y}) \geq 2d_1 \quad (3)$$
- From triangle inequality, we have

$$d(\mathbf{x}, \mathbf{y}) \geq d(\mathbf{x}, \mathbf{z}) - d(\mathbf{y}, \mathbf{z})$$

$$w(\mathbf{x} + \mathbf{y}) \geq w(\mathbf{x} + \mathbf{z}) - w(\mathbf{y} + \mathbf{z})$$

this expression. It's Hamming weight between \mathbf{u} and \mathbf{u} prime

(Refer Slide Time 37:22)



plus Hamming weight of u plus u prime plus

(Refer Slide Time 37:26)

Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is
$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = \mathbf{u}|\mathbf{u} + \mathbf{v}|$ and $\mathbf{y} = \mathbf{u}'|\mathbf{u}' + \mathbf{v}'|$ be two distinct codewords in C .
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}')$$
where $w(\mathbf{z})$ is the Hamming weight of \mathbf{z} .
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case
$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}')$$
- Since $\mathbf{u} + \mathbf{u}'$ is a nonzero codeword in C_1 , $w(\mathbf{u} + \mathbf{u}') \geq d_1$. Therefore
$$d(\mathbf{x}, \mathbf{y}) \geq 2d_1 \quad (3)$$
- From triangle inequality, we have
$$d(\mathbf{x}, \mathbf{y}) \geq d(\mathbf{x}, \mathbf{z}) - d(\mathbf{y}, \mathbf{z})$$
$$w(\mathbf{x} + \mathbf{y}) \geq w(\mathbf{x} + \mathbf{z}) - w(\mathbf{y} + \mathbf{z})$$

v plus v prime. And what we did just now is we lower bounded this. So then Hamming distance between x and y

(Refer Slide Time 37:40)

Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get

$$w(\underline{u + u' + v + v'}) \geq w(v + v') - w(u + u')$$

can be, so

(Refer Slide Time 37:42)

Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get

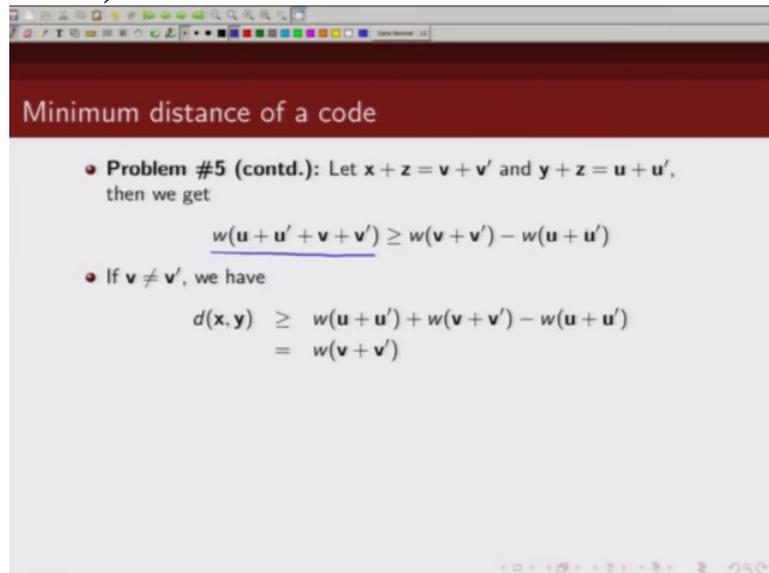
$$w(u + u' + v + v') \geq w(v + v') - w(u + u')$$

- If $v \neq v'$, we have

$$\begin{aligned} d(x, y) &\geq w(u + u') + w(v + v') - w(u + u') \\ &= w(v + v') \end{aligned}$$

this basically this we lower bounded by this quantity.

(Refer Slide Time 37:45)

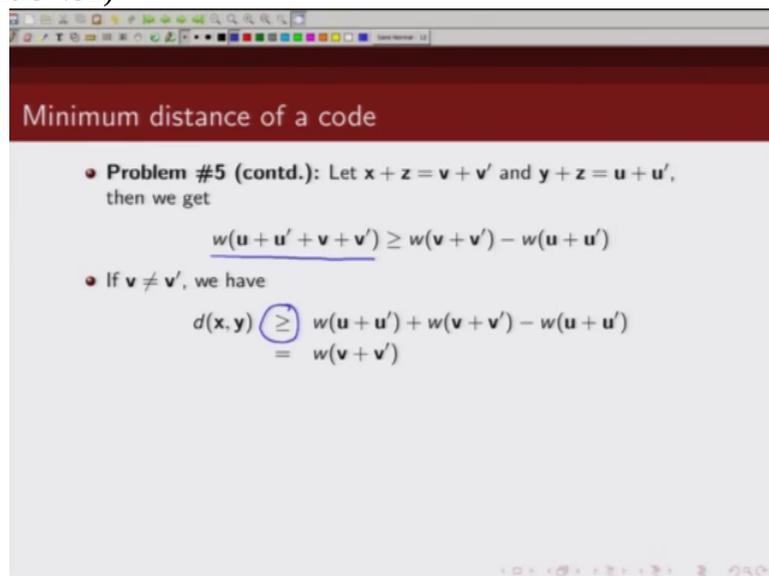


Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get
$$w(u + u' + v + v') \geq w(v + v') - w(u + u')$$
- If $v \neq v'$, we have
$$d(x, y) \geq w(u + u') + w(v + v') - w(u + u') = w(v + v')$$

If we plug that in here, what we get here is greater than equal to.

(Refer Slide Time 37:51)

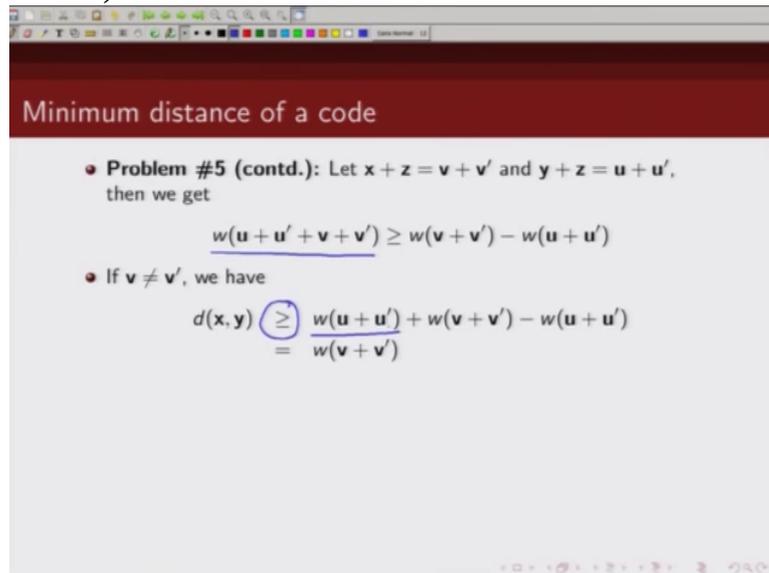


Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get
$$w(u + u' + v + v') \geq w(v + v') - w(u + u')$$
- If $v \neq v'$, we have
$$d(x, y) \geq w(u + u') + w(v + v') - w(u + u') = w(v + v')$$

So what we can write is the Hamming distance between x and y is then greater than or equal to this term comes

(Refer Slide Time 38:02)



Minimum distance of a code

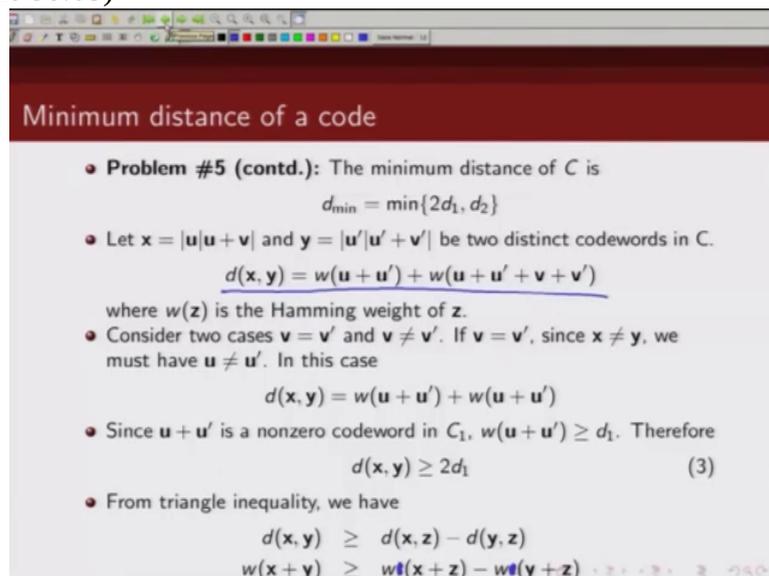
- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get

$$w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}') \geq w(\mathbf{v} + \mathbf{v}') - w(\mathbf{u} + \mathbf{u}')$$
- If $\mathbf{v} \neq \mathbf{v}'$, we have

$$d(\mathbf{x}, \mathbf{y}) \geq \frac{w(\mathbf{u} + \mathbf{u}') + w(\mathbf{v} + \mathbf{v}') - w(\mathbf{u} + \mathbf{u}')}{w(\mathbf{v} + \mathbf{v}')} = w(\mathbf{v} + \mathbf{v}')$$

from here, this term

(Refer Slide Time 38:05)



Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = |\mathbf{u}|\mathbf{u} + \mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'|\mathbf{u}' + \mathbf{v}'|$ be two distinct codewords in C .

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}')$$
 where $w(\mathbf{z})$ is the Hamming weight of \mathbf{z} .
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}')$$
- Since $\mathbf{u} + \mathbf{u}'$ is a nonzero codeword in C_1 , $w(\mathbf{u} + \mathbf{u}') \geq d_1$. Therefore

$$d(\mathbf{x}, \mathbf{y}) \geq 2d_1 \quad (3)$$
- From triangle inequality, we have

$$d(\mathbf{x}, \mathbf{y}) \geq d(\mathbf{x}, \mathbf{z}) - d(\mathbf{y}, \mathbf{z})$$

$$w(\mathbf{x} + \mathbf{y}) \geq w(\mathbf{x} + \mathbf{z}) - w(\mathbf{y} + \mathbf{z})$$

and this term is lower bounded by this term

(Refer Slide Time 38:09)

Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get

$$w(\underline{u + u' + v + v'}) \geq w(v + v') - w(u + u')$$

here. So we write it

(Refer Slide Time 38:12)

Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get

$$w(\underline{u + u' + v + v'}) \geq w(v + v') - w(u + u')$$

- If $v \neq v'$, we have

$$d(x, y) \begin{matrix} \geq \\ = \end{matrix} \frac{w(u + u') + w(v + v') - w(u + u')}{w(v + v')}$$

here, now this can be further written as Hamming weight of v plus v prime because these two cancel out. So what

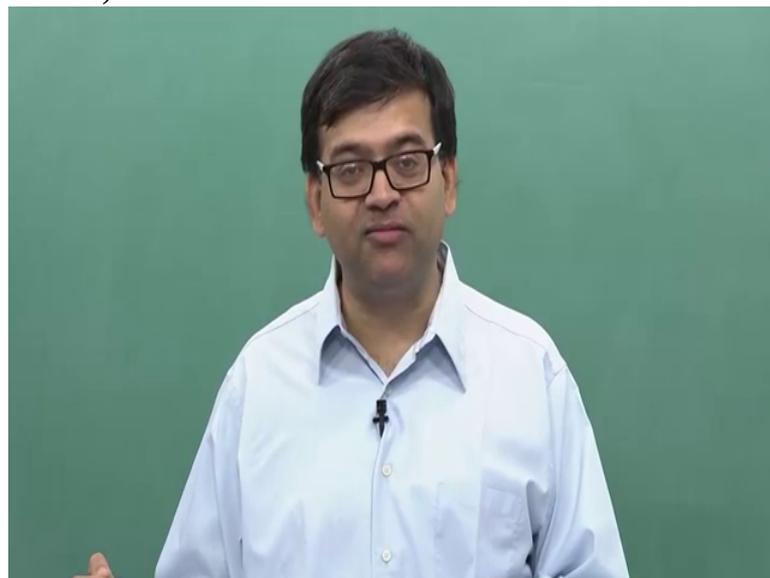
(Refer Slide Time 38:23)

Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get
$$w(u + u' + v + v') \geq w(v + v') - w(u + u')$$
- If $v \neq v'$, we have
$$d(x, y) \geq w(v + v') - w(u + u') + w(u + u') = w(v + v')$$

we have shown is when v is not same as v' the Hamming distance between

(Refer Slide Time 38:30)



x and y is greater than equal to Hamming weight

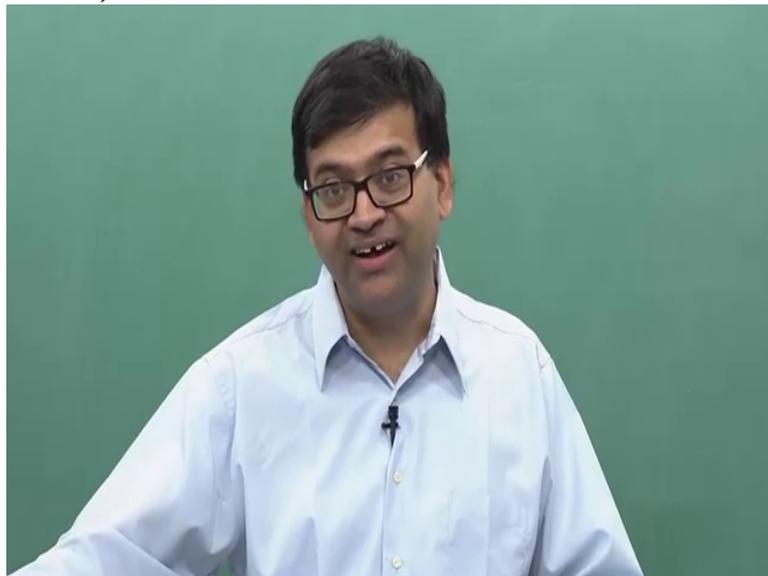
(Refer Slide Time 38:36)

Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get
$$w(u + u' + v + v') \geq w(v + v') - w(u + u')$$
- If $v \neq v'$, we have
$$d(x, y) \geq \cancel{w(u + u')} + w(v + v') - \cancel{w(u + u')} = w(v + v')$$

of v plus v' . And is v plus v' prime? v and v' are valid codewords in linear block code C with minimum distance d so v plus v' will be another valid codeword in C

(Refer Slide Time 38:53)



whose minimum distance is d . So then we can

(Refer Slide Time 38:58)

Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get
$$w(u + u' + v + v') \geq w(v + v') - w(u + u')$$
- If $v \neq v'$, we have
$$d(x, y) \geq w(u + u') + w(v + v') - w(u + u') = w(v + v')$$
- Since $v + v'$ is a nonzero codeword in C_2 , $w(v + v') \geq d_2$, we have
$$d(x, y) \geq d_2 \quad (4)$$

write this as Hamming distance between x and y is greater than equal to d 2. So we look, comparing equation number 4 and equation

(Refer Slide Time 39:11)

Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get
$$w(u + u' + v + v') \geq w(v + v') - w(u + u')$$
- If $v \neq v'$, we have
$$d(x, y) \geq w(v + v')$$

number 3,

(Refer Slide Time 39:12)

Minimum distance of a code

- **Problem #5 (contd.):** The minimum distance of C is

$$d_{\min} = \min\{2d_1, d_2\}$$
- Let $\mathbf{x} = |\mathbf{u}| \mathbf{u} + \mathbf{v}|$ and $\mathbf{y} = |\mathbf{u}'| \mathbf{u}' + \mathbf{v}'|$ be two distinct codewords in C .

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}')$$
 where $w(\mathbf{z})$ is the Hamming weight of \mathbf{z} .
- Consider two cases $\mathbf{v} = \mathbf{v}'$ and $\mathbf{v} \neq \mathbf{v}'$. If $\mathbf{v} = \mathbf{v}'$, since $\mathbf{x} \neq \mathbf{y}$, we must have $\mathbf{u} \neq \mathbf{u}'$. In this case

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{u} + \mathbf{u}') + w(\mathbf{u} + \mathbf{u}')$$
- Since $\mathbf{u} + \mathbf{u}'$ is a nonzero codeword in C_1 , $w(\mathbf{u} + \mathbf{u}') \geq d_1$. Therefore

$$d(\mathbf{x}, \mathbf{y}) \geq 2d_1 \quad (3)$$
- From triangle inequality, we have

$$d(\mathbf{x}, \mathbf{y}) \geq d(\mathbf{x}, \mathbf{z}) - d(\mathbf{y}, \mathbf{z})$$

$$w(\mathbf{x} + \mathbf{y}) \geq w(\mathbf{x} + \mathbf{z}) - w(\mathbf{y} + \mathbf{z})$$

if we compare these two equations we can write that minimum distance of the code is minimum of $2d_1$ or d_2 , Ok.

(Refer Slide Time 39:28)

Minimum distance of a code

- **Problem #5 (contd.):** Let $\mathbf{x} + \mathbf{z} = \mathbf{v} + \mathbf{v}'$ and $\mathbf{y} + \mathbf{z} = \mathbf{u} + \mathbf{u}'$, then we get

$$w(\mathbf{u} + \mathbf{u}' + \mathbf{v} + \mathbf{v}') \geq w(\mathbf{v} + \mathbf{v}') - w(\mathbf{u} + \mathbf{u}')$$
- If $\mathbf{v} \neq \mathbf{v}'$, we have

$$d(\mathbf{x}, \mathbf{y}) \geq w(\mathbf{u} + \mathbf{u}') + w(\mathbf{v} + \mathbf{v}') - w(\mathbf{u} + \mathbf{u}') = w(\mathbf{v} + \mathbf{v}')$$
- Since $\mathbf{v} + \mathbf{v}'$ is a nonzero codeword in C_2 , $w(\mathbf{v} + \mathbf{v}') \geq d_2$, we have

$$d(\mathbf{x}, \mathbf{y}) \geq d_2 \quad (4)$$
- From (3) and (4) we have

$$d(\mathbf{x}, \mathbf{y}) \geq \min\{2d_1, d_2\}$$

Now let us show that

(Refer Slide Time 39:31)

Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get
$$w(u + u' + v + v') \geq w(v + v') - w(u + u')$$
- If $v \neq v'$, we have
$$d(x, y) \geq w(u + u') + w(v + v') - w(u + u') = w(v + v')$$
- Since $v + v'$ is a nonzero codeword in C_2 , $w(v + v') \geq d_2$, we have
$$d(x, y) \geq d_2 \quad (4)$$
- From (3) and (4) we have
$$\underline{d(x, y) \geq \min \{2d_1, d_2\}}$$

there exists a code

(Refer Slide Time 39:33)



with, minimum distance of code is indeed

(Refer Slide Time 39:36)

Minimum distance of a code

- **Problem #5 (contd.):** Let $x + z = v + v'$ and $y + z = u + u'$, then we get
$$w(u + u' + v + v') \geq w(v + v') - w(u + u')$$
- If $v \neq v'$, we have
$$d(x, y) \geq w(u + u') + w(v + v') - w(u + u') = w(v + v')$$
- Since $v + v'$ is a nonzero codeword in C_2 , $w(v + v') \geq d_2$, we have
$$d(x, y) \geq d_2 \quad (4)$$
- From (3) and (4) we have
$$d(x, y) \geq \min \{2d_1, d_2\}$$

equal minimum of $2d_1$ or d_2 . So let us take two minimum weight codewords

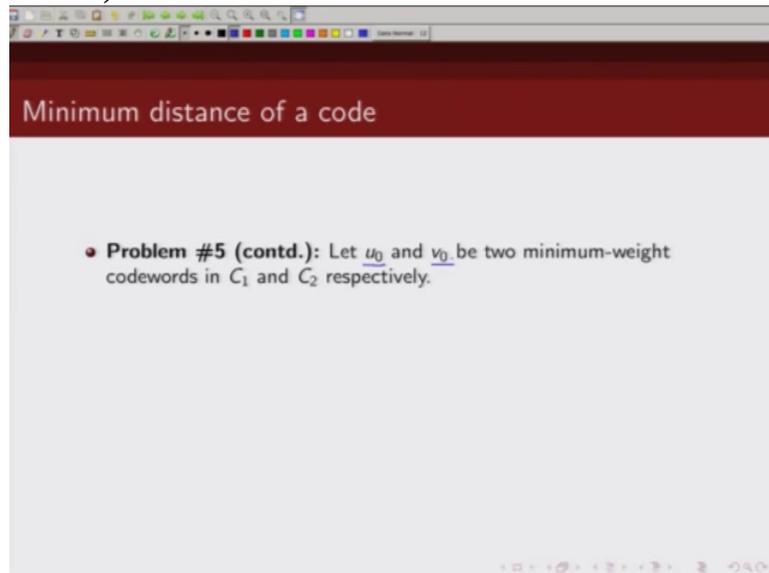
(Refer Slide Time 39:44)

Minimum distance of a code

- **Problem #5 (contd.):** Let u_0 and v_0 be two minimum-weight codewords in C_1 and C_2 respectively.

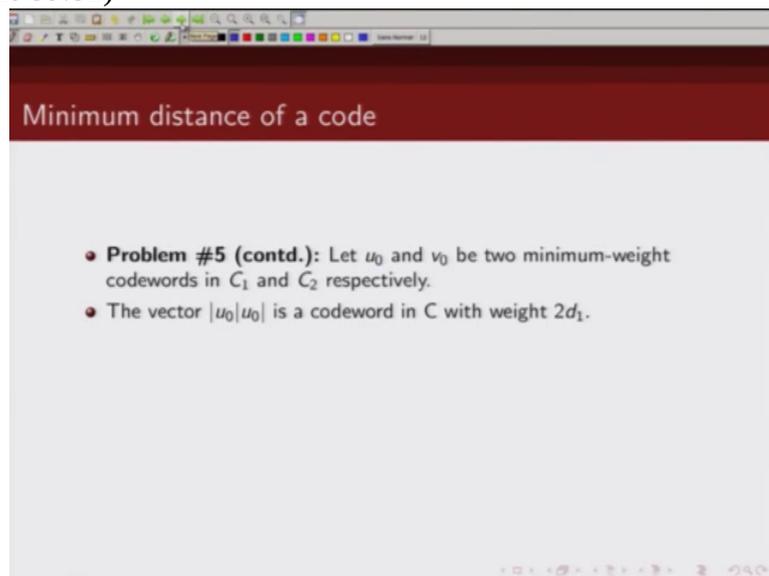
in C_1 and C_2 . Let us call them u_0

(Refer Slide Time 39:49)



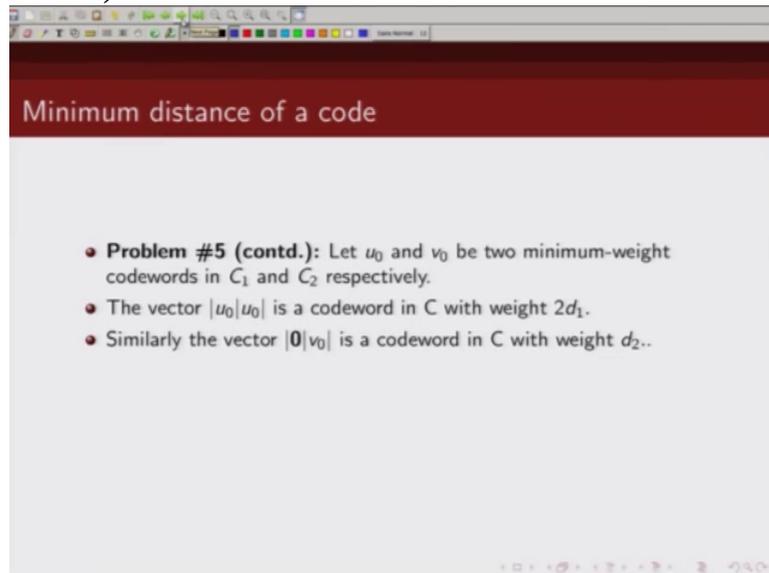
and v_0 .

(Refer Slide Time 39:52)



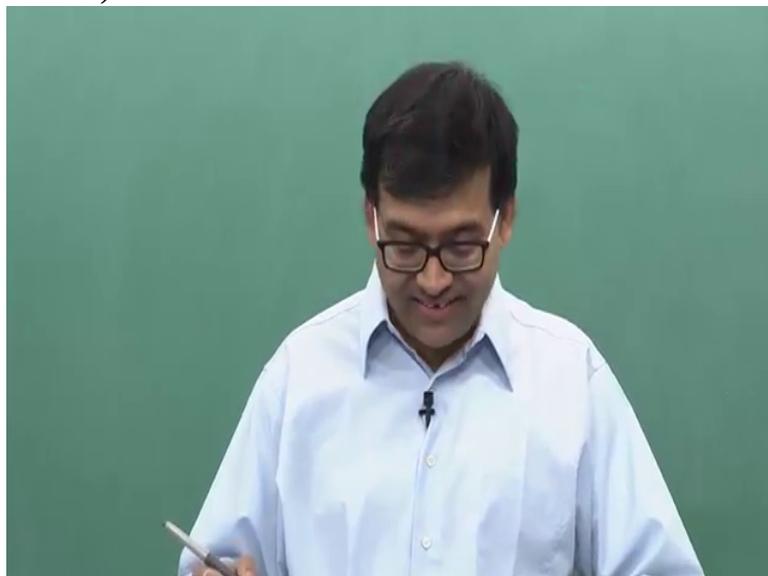
Now this is a valid codeword in C and what's its minimum distance? It is 2 times d_1 .

(Refer Slide Time 40:14)



So if we take v_0 to be all zero codeword, what we get is $|u_0|0$ and $|u_0|0$, this is a valid codeword in C . And its minimum distance is 2 times d_1 . Similarly if we take u_0 to be all zero codeword, then what we get is $|0|v_0$ and $|0|v_0$, whose minimum distance is d_2 . So hence we have shown that there,

(Refer Slide Time 40:28)



basically there minimum distance of code, of this code new code C is indeed minimum of $2d_1$ or d_2 ,

(Refer Slide Time 40:40)

Minimum distance of a code

- **Problem #5 (contd.):** Let u_0 and v_0 be two minimum-weight codewords in C_1 and C_2 respectively.
- The vector $|u_0|u_0|$ is a codeword in C with weight $2d_1$.
- Similarly the vector $|0|v_0|$ is a codeword in C with weight d_2 .
- Therefore

$$d(\mathbf{x}, \mathbf{y}) = \min \{2d_1, d_2\}$$

Ok. Thank you