SORTING AND SEARCHING – 20 QUESTIONS GAME - 08

Hello guys in last video we saw the linear search we saw how inefficient it is to search a very big number in a very big array suppose you have million of numbers in a array and you want to search close to million then it will take around a million of steps to search that number in liner search so it is very inefficient because you do not have that much of memory power or that much of even time to spend on an array of size one million and that too if the array is sorted so here what we are going to see is the very very good algorithm known as binary search whenever you get an array you sort it either in ascending order or in descending order, you are getting some extra information which you can used to search the search any element in an array in very less number of steps that exactly what we are going to see here so in the game, in the game that you saw in the video there were two players one was ravi and amit they were playing a game so ravi has an idea that he can guess a the number which amit has thought in a very less amount of iterations so the game is like this you search you guess a number between one to thousand and then the second person asks second person first computes the midpoint of that array so in one to thousand the midpoint is five hundred then he asks whether this whether the number that you have guessed is equal to five hundred or not if it is five hundred then ok i got the number if it is not five hundred then it is he ask whether it is less than five hundred or it is greater than five hundred so suppose the number you have guessed is less than five hundred then you can see that you can safely discard the other five hundred elements in the array because your number will never go beyond five hundred you have already said it is less than five hundred and the numbers are sorted so you can never go you do not have to search the other part of the array so you can safely discard the numbers from five hundred to thousand so number of elements that you have left now is one to five hundred then again the i will ask i will compute the midpoint of this array which is left that is one to five hundred which is the midpoint is two fifty then you will then i will ask you again whether your number is two fifty if you says yes then ok if you says no then again i will ask whether it is less than two fifty or more than two fifty, if it is more than two fifty then again same logic i will use since it is more than two fifty it i do not have to search it less than to fifty because it will never happen that the element will present in the array between one to two fifty so i can safely discard the numbers from one to two fifty now the arrays which i have left is two fifty to five hundred again i will ask the same question that is i will compute that midpoint of two fifty to five hundred and i will use that midpoint to ask the question and go to the left part of the array and right part of the array this is the logic of binary system in this you are a every each iteration you are whenever you are computing the midpoint you are discarding the half part of the array this way you are size of the array is reducing by two at each iteration so you will exactly use this logic to do the binary search so let me right the definition of the binary search. Def binary search it will take two arguments one is the array let's say array name is 'a' and other is the element that i want to search great, now i will create two variables i will use this variables to position my array, i want to know i will create a variable let's say first position first pos which is zero because array starts with zero so i can safely take zero as my first position and the last position last pos sorry which is

the last position of the array, python is an function named it as len through which you can compute the length of an array so suppose if i type len of a i will get the length of array now it gives the length of array gives the number of elements in the array but we know that list or array whatever you said in python starts with zero so you have to take minus one so let me try this in console suppose my array name is array one and the elements are one comma two comma three comma four comma five ok now if you type length of array one you will get oh there are five elements so you will get five but the position of fifth element is what? Is actually four because the array starts with zero, zero one two three four so suppose you want to retrieve the last element will type array sorry array one and if you type four here you will get the fifth element or you can type array length of this array one ok this length of array one will return the number of elements in array one which is five so number of elements in array one is five if i put minus one here so this this whole thing means it is five which is return by the length of array one minus one which is four so this will print give me the fifth element fourth element which is five so this is exactly what we are going to do here so i will put the last position is length of array minus one ok also i will take a flag value which i will use to see whether i found my element or not ok this warning means that all this variables have been you have created this variables but you are not using it so as, as soon as i use it all this warnings will go ok now i will create a while loop let's see how to do this? And let me write this first and i will explain it how what exactly i am trying to do. So i will type first pos is less than last pos and flag is equal equal to zero what is this mean that? Keep this loop going on until the first position is less than the last position which is this two variables, if first position is always less than last position and flag is equal equal to zero i told you flag you can let me write a comment here flag means that flag go to zero that element is not been found so it means that you should continue the loop until first position is less than last position and you haven't found the element you are looking for till than you have to continue this loop ok so what i will do first what was the strategy, so first i will compute the midpoint of the array so i will compute mid is equal to you write so i can compute the know to compute the midpoint of an array i will take the first element the position of first element i mean the first position of the array and the last position of the array and i will add it and divide it by two this why i can iteratively i can compute the midpoint of any sub array so i will write first pos plus last pos divided by two now if i write divide by two it will return me a float value, you can check here suppose if i five plus two divide by two it is giving me three point five but i exactly want a integer value so suppose the number is odd it should give a integer value either left or right i don't care much but it should give me a integer position because it is a position if i use this float value then i will not be able to retrieve the element at that position so i need an integer value for that i will do the int there is called as integer division in python that is i will write five plus two, to this division it will give me three this is what i want so i will put this one more division operator here and i am done so i computed the mid now what i will check? i will check whether this element 'x' is present in mid or not i will write if 'x' is equal equal to a of because made is of position if it is there then i am done, i have to check for another thing so i will write i will make the flag is equal to one it means i found the element and i will write print element present at position i can write i can give the position of this mid variable which is i can write str mid you can add plus one also so that you can get the exact position but i am not writing it because i am assuming that you all know that array starts with zero so in that

context i am just printing the mid value that's it you can return from here, it means that whenever you found the whenever you get the element which you are looking for you can return it you do not have to other computation i will just return from this, this return will return the function value which i am not returning anything it means it will get out from the function to the main i am calling the function that's it ok now if it is not present at the mid position then i have to check whether it is on the left side or right side, for that i will write else if it is not present at the mid position if this 'x' is less than array of mid which means if it if 'x' is less than mid i can discard the right side of the array which means what? I can safely write that last position is shifted to what? Mid minus one because mid and after mid i can discard because mid is i the first question in the first condition i have checked whether it is present at mid or not, if it is not present at mid then i can discard the mid and all the elements after the mid so i will safely i will reduce the size of my array by positioning the last position to mid minus one this why i am discarding the other elements of the array or if it is not wait sorry if it is not less than the mid it is more than the mid then i can discard the first part of the array which is from zero to mid i can discard for that i will just shift my first position has first pos is equal to mid plus one that's it ok and then the programme is over and if the function doesn't returned anything after all these things it means the array i didn't find the value which i am looking for so i can safely print here after this loop it means that loop is over but still i am not able i didn't find the element i will write the number is not present that's it now in last function the linear search you were printing the number of iterations also how many iterations it takes so here also will i would like to print the number of iteration it is taking to find this value 'x' so i will create a count variable count assign it to zero and at every iteration of while loop i will increase the value now here i have printed the element where it is present i will just print that the number of iterations are which is string of count ok cool, i just need to call this function so in order to call this function here i will create an array so let's say my array name is 'a' and i will enter the elements in this array for i in range one to five hundred let's say a dot append i and i will call this function which is binary search and i have to give the and let's say the number i have to search is seventy so our programme is complete so i will just run this program to see the output. Ok that is showing the number is not present and i exactly did what the logic was present for the binary search and i know that number is there suppose if i type 'a' you can see that seventy which i am looking for is present but still it is showing the element is not present, let me try this for small number lets say number of elements or ten what i want to search is four let me try this, still it is showing that number is not present let me see the numbers one two three four five six seven eight nine ten, and four is there now what is exactly happening is that whenever the condition that i put here while first position is less than last position it might happen that some time the first position and last position will be the one number let say first position and last position are pointing at four at that time still we have one iteration search is required but this condition is will fail that is first position is less than last position it is showing that here the first position is not less than last position it is equal to last position and here it will fill and i will not be able to search this element so to do this i have to make this condition as less then equal to which means when first position is less than equal to last position until then you have to search for the element now you can see if i run this, it will show that number is present at position three zero one two three and the number of iterations is four because four times i have to switch to this, now

let’s see this, number of elements will be hundred and one and then i will search for number seventy and run this see the number is present at sixty nine, seventy. Now in linear search what we were looking for, we search for a number the number of elements we put where thousand ok ok and we were searching for the number thousand and it was taking thousand of iteration now here let’s see how many times, how many iterations it will take? I will run this the see the number is present at position nine nine nine and in ten iterations i was able to search this number ten iterations where it was taking thousand iteration in linear search it just takes ten iteration in binary search, suppose i make this one million let’s say ok and i want to search this number let’s say one lack ok see how many iteration it will take in just twenty iteration, in just twenty iteration i was able to find the number, whether as you all know with to find this number it would have taken one lack iteration in linear search so this is the beauty of binary search, you can do actually it takes exactly log of the number of elements in the array that is if the number of elements let’s say sixteen then it will take four iteration and thirty two then five iteration and sixty four six iteration so on because every time you are dividing by two and discarding the rest other part, you are taken only one part at a time this way one half of the array is every time discarded and you are reducing your space by two exactly by two every time by two by two by two by two becomes log of two log of base two so yeah we are finished with the binary search please if you have any questions post it on discussion form or you can if you think you can, this is a very simple programme you think you can create a more efficient more elegant program like this better than this please post it on discussion form it might help other people to learn ok guys thank you.