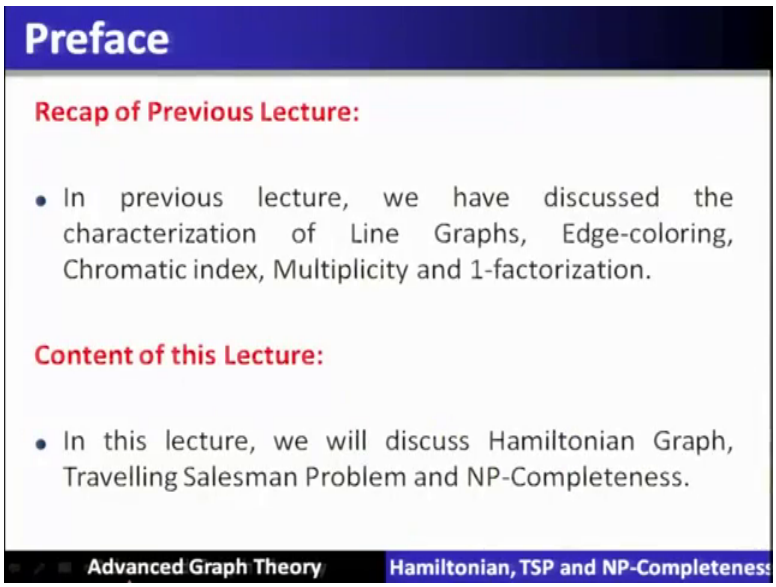
Advanced Graph Theory
Prof. Rajiv Misra
Department of Computer Science and Engineering
Indian Institute of Technology, Patna

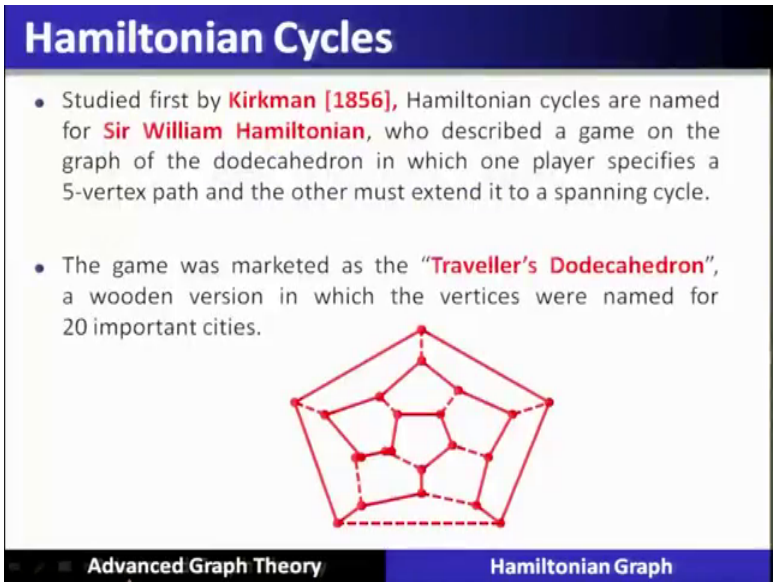
Lecture – 22
Hamiltonian Graph, TSP and NP-Completeness

(Refer Slide Time: 00:19)



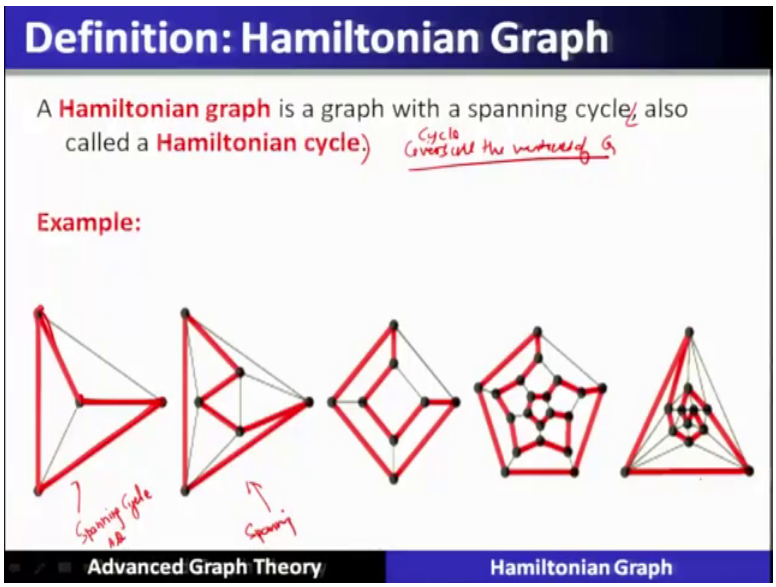
Hamiltonian graph, travelling salesman problem and NP-Completeness; recap of previous lecture, we have discussed the characterization of line graphs, edge-coloring, chromatic index, multiplicity and 1-factorizations. Content of this lecture; we will discuss Hamiltonian graph, travelling salesman problem and NP-Completeness.

(Refer Slide Time: 00:45)



Hamiltonian graph; it was studied first by Kirkman in 1856, Hamiltonian cycles are named after Sir William Hamiltonian, who described the game on the graph of the dodecahedron in which one pair specifies a 5-vertex path and the other player must extend it to a spanning cycle. The game was marketed as the Traveller's Dodecahedron; a wooden version in which a vertices were named for 20 important cities.

(Refer Slide Time: 01:23)



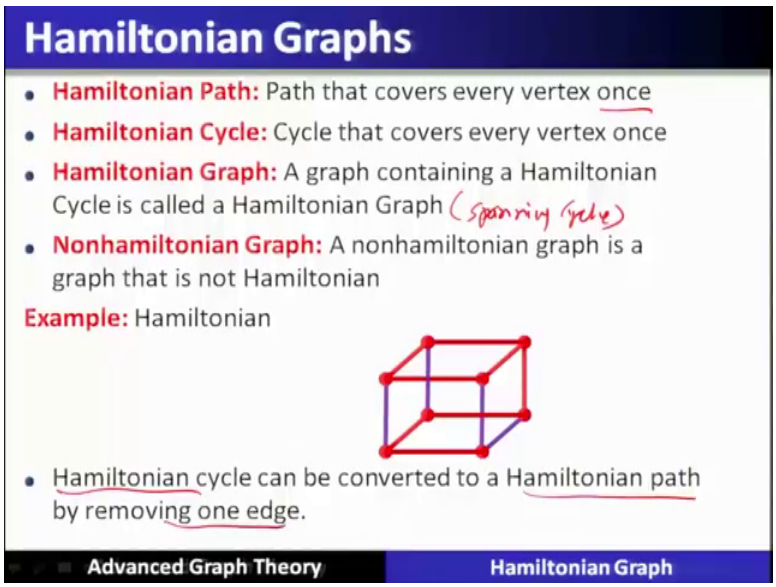
The definition of Hamiltonian graph; Hamiltonian graph is a graph with a spanning cycle also called as the Hamiltonian cycle. So, let us see in this particular example, the

spanning cycle of this particular graph is shown in the red color, it will it is spans across all the vertices, this particular cycle is called a spanning cycle which passes through all the or which covers all the vertices of the graph G that is the cycle.

So, the graph which contains this cycle spanning cycle is called Hamiltonian graph. Another example; here we can see this side particular cycle; spanning cycle exists in another graph which is spans through or which covers all the vertices of this particular graph.

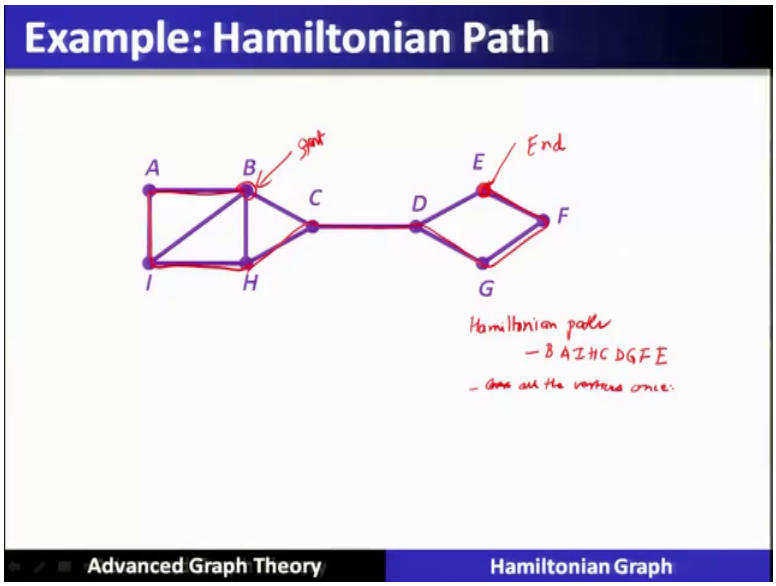
Similarly, all the examples which are shown over here are the example of Hamiltonian graphs because that contains a spanning cycle within it.

(Refer Slide Time: 02:42)



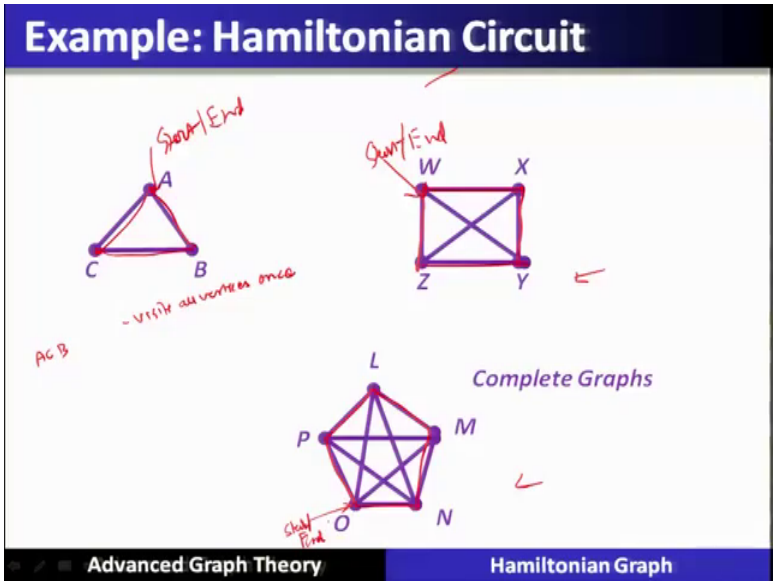
Hamiltonian graphs; Hamiltonian path is the path that covers every vertex. Now see, once it is may not be required because path contains the vertices exactly once. So, if a path that covers all the vertices exactly once, then it is called Hamiltonian path. Hamiltonian cycle is a cycle that covers every vertex once. Hamiltonian graph a graph containing the Hamiltonian cycle is called a Hamiltonian graph or we can also say a graph having a spanning cycle is called a Hamiltonian graph. A non Hamiltonian graph; a non Hamiltonian graph is a graph that is not Hamiltonian; that means, that does not contains i spanning cycle or there is no Hamiltonian cycle present in the graph they are called non Hamiltonian graphs.

(Refer Slide Time: 03:51)



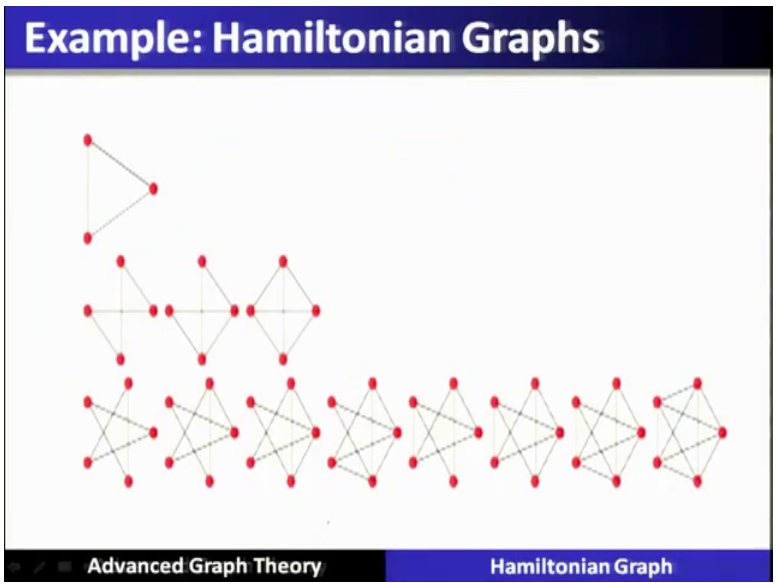
Example of a Hamiltonian graph is shown over here. So, Hamiltonian cycle can be converted it to a Hamiltonian path just by removing one edge from it here the example we can see the example of a Hamiltonian path. So, if we start from vertex B covers through C, then D and n set vertex C. So, it will start at B and it will finish at at E, the path is called Hamiltonian path, why because it covers all the vertices exactly, once Hamiltonian path which we have identified here is B A I H C D G F E. So, here this particular path has touched all the vertices; that means, it covers all the vertices once, hence it is called Hamiltonian path.

(Refer Slide Time: 05:02)



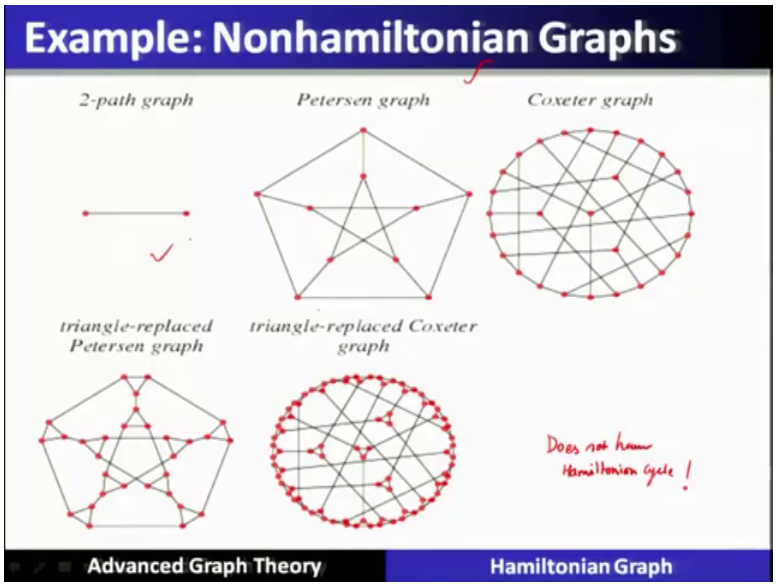
Here, it is having a Hamiltonian circuit, let us see this the example, we start at a visit C, then visit B and then end at to the same points, it is called a circuit and it visits all the vertices once; that is ACB, we have identified, this is called Hamiltonian circuit. Similarly, other two graphs; we can also identify, let us start from W visit Z visit Y X and W will start and end at the same points, hence, it is having a circuit which visits all the vertices exactly once called Hamiltonian circuit. Let us take this example also, we start from here and come back at the same points start and end at the same points and visit all the vertices exactly once called Hamiltonian circuit.

(Refer Slide Time: 06:19)



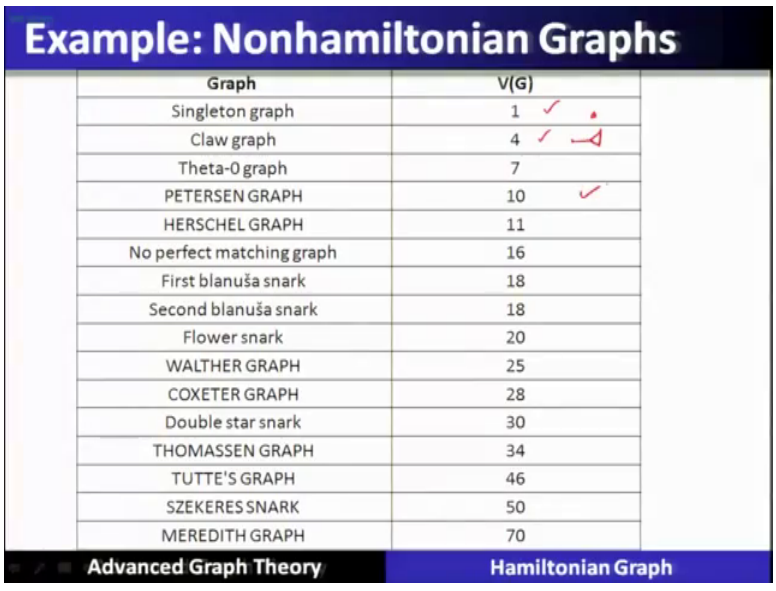
Similarly, these are different graph shown which are basically the Hamiltonian graphs having the Hamiltonian cycle or a circuit within it, these are the example which is which are non Hamiltonian graph why because it does not have the Hamiltonian circuit or a Hamiltonian cycle.

(Refer Slide Time: 06:33)



So, it does not have Hamiltonian cycle therefore, they are all non Hamiltonian graphs. So, for example, here in Petersen graph, there is no cycle possible why because these end vertices having degree one, this is having Petersen graph is another example which does not have the Hamiltonian cycle. Hence, this is not Hamiltonian graph.

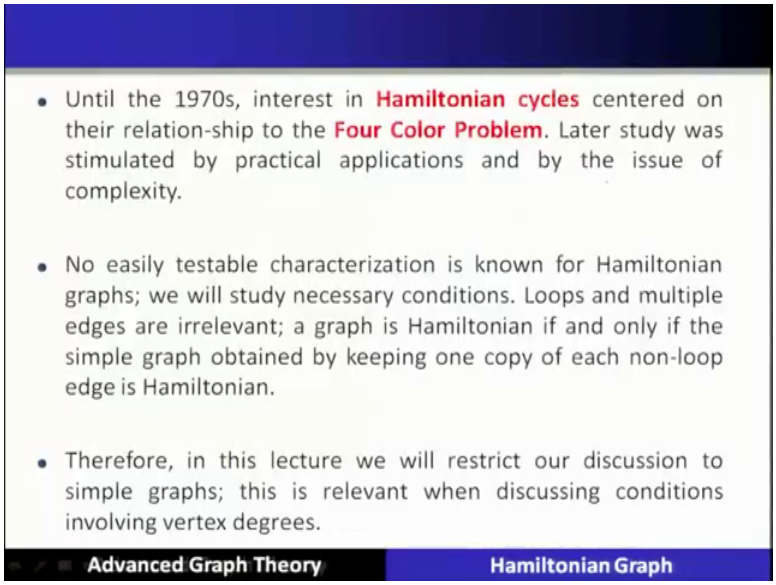
(Refer Slide Time: 07:21)



Similarly, all other graphs which does not have the Hamiltonian cycle within it, so, non Hamiltonian graph example are shown over here that is the graph with 1 vertex, then claw graph is also a graph with a 4 vertices. So, this is a claw graph, this also non

Hamiltonian. Similarly, all other graphs; Petersen, we have already seen that is non Hamiltonian.

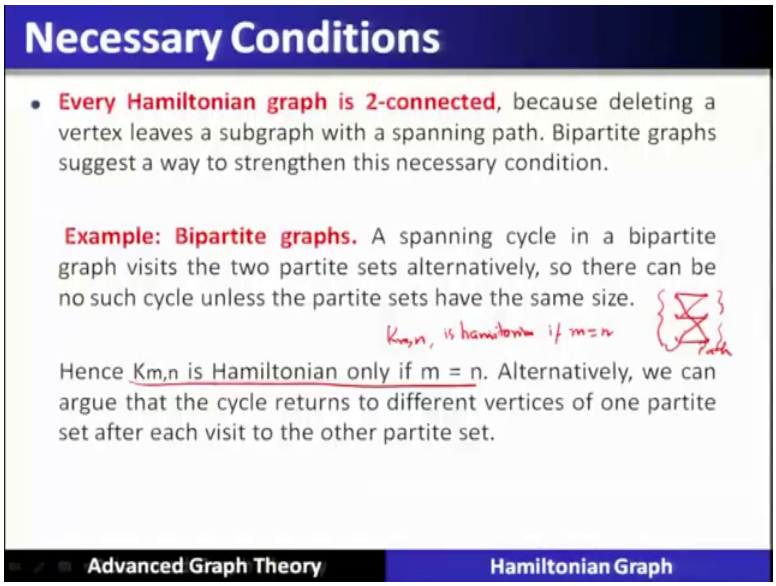
(Refer Slide Time: 07:44)



So, until 1970, interest in Hamiltonian cycle centered on their relationship to the 4 color problem, later study was stimulated by practical applications and by the issue of complexity. so, because of this issue of complexity this Hamiltonian cycles are going to be an important for analysis of the algorithms.

No easily testable characterizations is known for Hamiltonian graphs, we will study the necessary conditions, here loops and multiple edges are irrelevant, a graph is Hamiltonian if and only if the simple graph obtained by keeping one copy of each non loop edge is Hamiltonian. Therefore, in this lecture, we will restrict our discussion to the simple graphs only. This is relevant when discussing the conditions involving the vertex degrees.

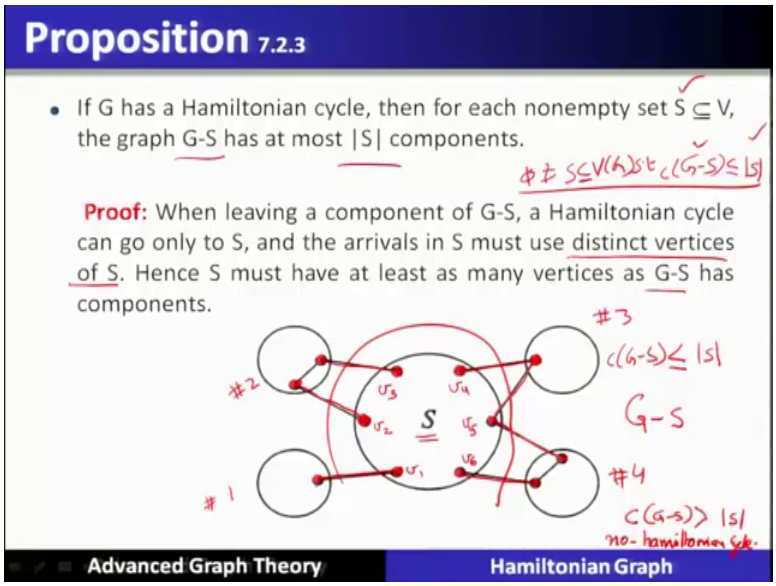
(Refer Slide Time: 08:40)



Necessity conditions for Hamiltonian graph; here we see that every Hamiltonian graph is 2-connected because deleting a vertex leaves the sub graph with by spanning path. So, we also called it as Hamiltonian path.

Now, bipartite graph suggest a way to strengthen this particular necessity conditions. So, as spanning cycle in a bipartite graph visits two partite sets alternatively. So, there can be no such cycle unless the partite sets have an equal size. So, take this particular example, this particular partite set has 3 elements, this is 2 elements. So, a spanning cycle in a bipartite graph visits 2 sides. So, there can be no such cycle unless bi partite sets is of same size. So, this becomes a path to complete a cycle, it has to have another vertex and then only a cycle is possible. So, the partite sets. So, bipartite graph K, m n is Hamiltonian only if if m is equal to n, alternatively, we can argue that the cycle returns to a different vertex of one partite sets as we have seen here will return to a different partite sets after each visits to the other partite sets.

(Refer Slide Time: 10:21)



Proposition; this particular proposition will establish the necessary conditions for a graph to be Hamiltonian or whether the Hamiltonian cycle is given in a graph or not. So, if G has a Hamiltonian cycle, then for every non empty set S which is a sub set of vertices of the graph G such that the graph without the that set S that is G minus S graph has at most the cardinality of S number of components. So, meaning to say that.

So, we if we take a subset S which is n not empty, this sub set is sub set from the vertex set of that particular graph such that if we remove this particular set of vertices from the graph the number of components which will be there should be at most the cardinality of S. So, again, let us see that is it stated in the proposition. So, if G is a Hamiltonian cycle, then for every subset S the graph which is obtained after removing that S that is G minus S has at most S number of components.

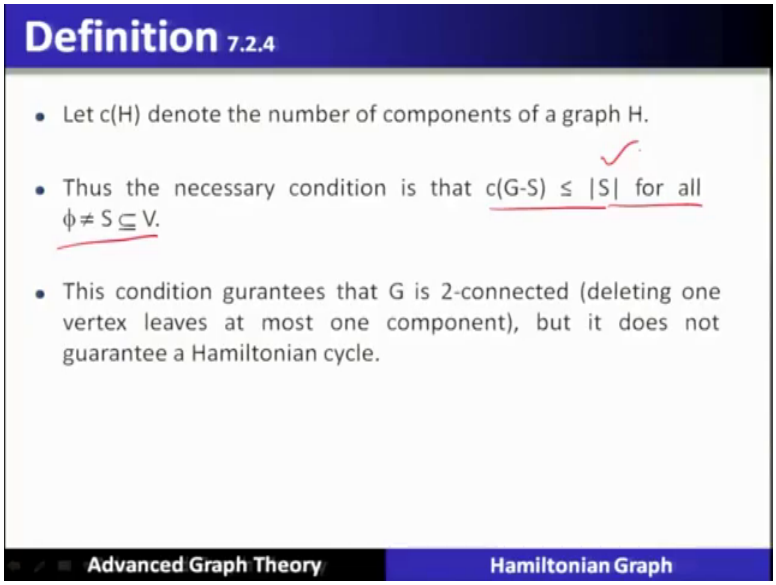
So; that means, the graph after removing that S will result in to the number of components and the number of components will be less than or equal to S that is the cardinality of S. So, let us see the proof. So, the proof says that when let us take this is G minus S and this is S. So, when this S will be removed, then they will be components; component number 1, component number 2, component number 3, component number 4, 4 components will be there. Now when leaving a components of G minus S, let us say it component number 1 when we leave a Hamiltonian cycle can go only to S so; that

means, a particular vertex in every component will basically arrive on S with a distinct vertex.

Let us say, this is v 1, another component if it leaves this particular component number 2, it will arrive here on another distinct vertex; that means, it is not arriving on v 1 will arrive on another distinct vertex v 2. Similarly, another vertex when it will leave the component, it will arrive on another distinct vertex, this particular vertex will arrive on a distinct vertex v 4, it may also arrive here. Similarly, this vertex will arrive on a vertex v 5 and this particular vertex will arrive v 6. So, when a component; when it leaves the component of G minus S, the Hamiltonian cycle can go only to S and the arrival of S must use the distinct vertices of S, hence, S must have at least as many vertices as G minus S has the components.

So, every component, if it is arriving in on S after leaving it, G minus S, it will arrive on distinct vertices. So, at least that many number of vertices should be there in S. Hence, this particular condition; so, the number of vertices; so, the number of components of G minus S should be basically at least S or we can also say that the number of components of G minus S, if it is greater than S, then there is no Hamiltonian cycle present in the graph according to this particular proposition which we have already proved.

(Refer Slide Time: 15:14)

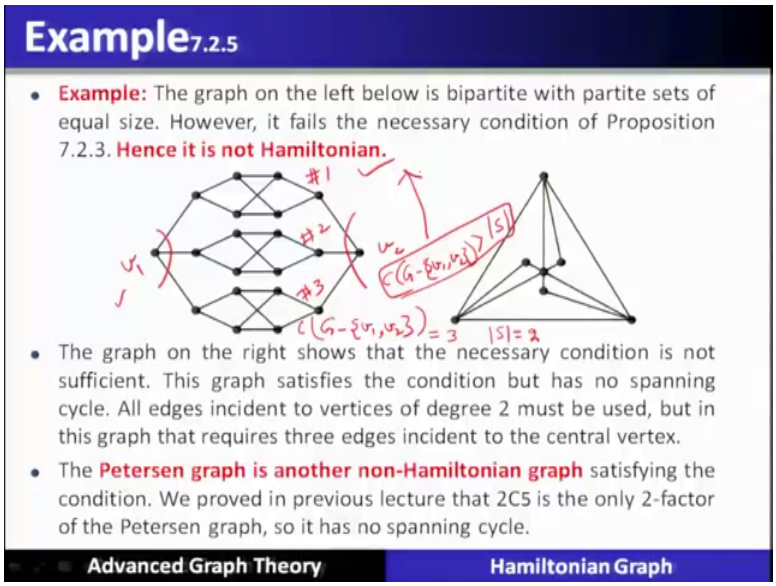


So, here in the definitions c of H denotes the number of components of a graph H. Thus a necessary conditions; I have already stated that the number of components in a graph G

minus S is basically at most S for all for all S which is the subset of V, so that means, the number of components should not exceed the number of elements present in S.

So, this condition guarantees that G is 2-connected that is deleting one vertex leaves at most one component, but it does not guarantees a Hamiltonian cycle.

(Refer Slide Time: 16:06)

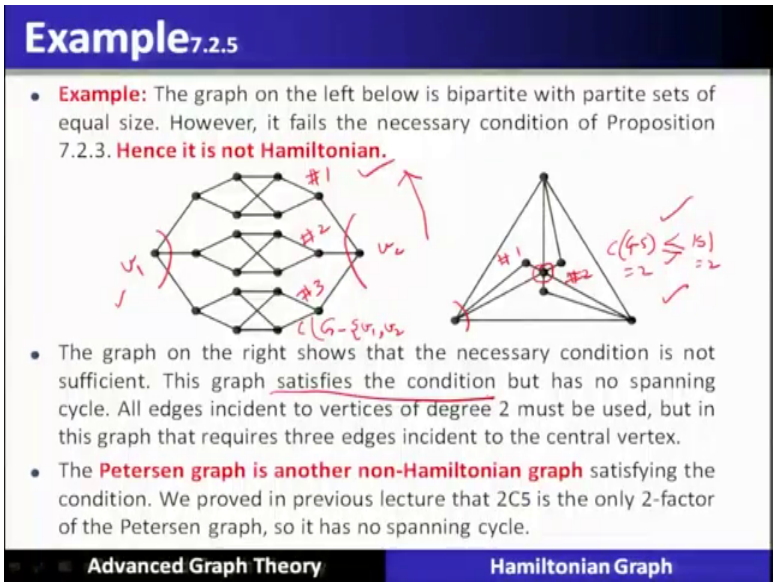


Because this is necessary condition, let us see the example to illustrate that proposition which is nothing but necessary conditions for the existence of a Hamiltonian cycle. So, in this particular graph which is nothing, but a bipartite graph, here we can see here that it is not Hamiltonian; that means, if you can find out a subset of vertices S in this particular graph and if we remove, then the resulting number of component will exceed the elements in S. Let us take this vertex v 1 and this vertex v 2 and remove it from the graph. So, the remaining graph G minus S; S consist of v 1 and v 2 will have component number 1, component number 2, component number 3, it will result in 3 different components.

So, the number of components in G minus S that is v 1 and v 2 will be 3 here in this case and cardinality of component v 2 is equal to 2. Hence the number of components C of G minus v 1, v 2 exceeds S which is nothing, but a violation of the necessary condition, therefore, this particular graph is not Hamiltonian; however, another graph which is shown over here on the right side this graph, if we consider, let us remove the 2 vertices. This is 1 vertex, if you remove; what will happen if you remove this vertex and; that

means, still these vertices will be connected and if we remove this vertex, then there will be some components the number of components again will be 2 because this will be one component.

(Refer Slide Time: 18:29)

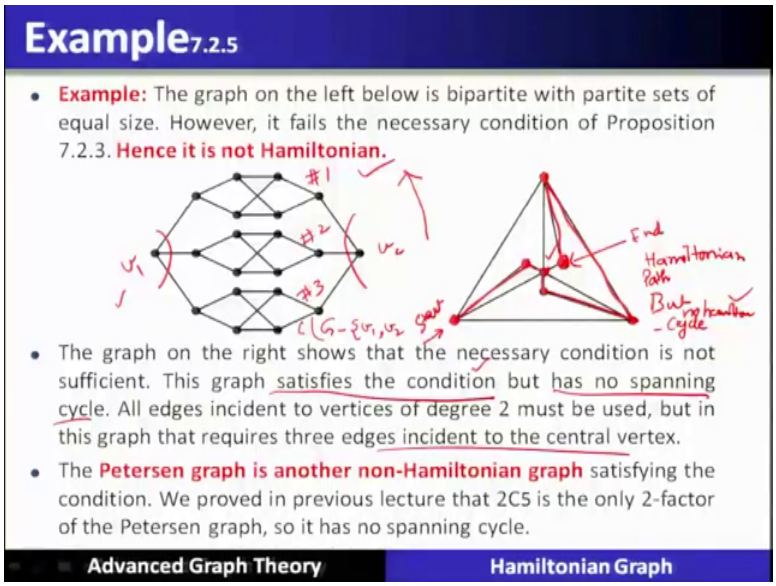


And remaining will be another component so; that means, if S is equal to 2 and the number of component after removing is also 2. So, equal to sign is basically satisfying so; that means, by the symmetry of graph, we can show that for all subsets this particular condition is satisfied. So, it satisfies the condition.

But the another question is whether it has the spanning cycle or a although, it satisfies the condition, but we will show that this particular necessary condition is not sufficient for the graph to have the Hamiltonian cycle.

Let us start by looking, if we start from this particular vertex and search by walking into the graph edges to find out a spanning cycle, let us start from this particular point go to the next vertex, then go to the next vertex and so on cover this particular vertex through that particular cycle or a walk and then come back to the same point. So, we start at this point and we finish at this point you see that.

(Refer Slide Time: 19:57)

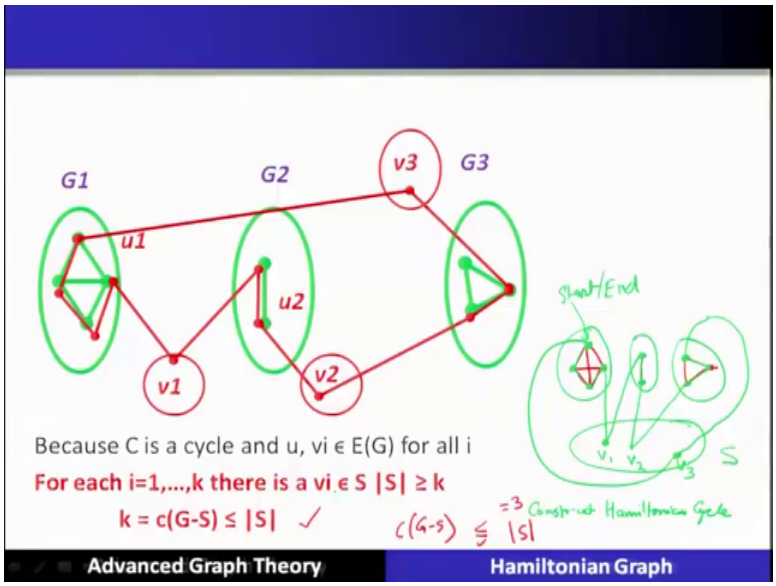


We have covered all the vertices. So, this is called a path because both the end points, we could not join from where we have started hence this particular path which is spans through all the vertices is called a Hamiltonian path.

But there is it is not a cycle or a circuit, but no Hamiltonian cycle present therefore, so, here it has no Hamiltonian spanning cycle, therefore, it is not that is the necessary condition is not sufficient for the existence of a Hamiltonian cycle that we have shown. So, all the edges incident to the vertices of degree 2 must be used, but in this graph that requires 3 edges incident to the central vertex and hence the necessary the condition is not sufficient. So, sufficient condition is also required for a graph to be Hamiltonian that we have already discussed.

Now, another important graph which is called a Petersen graph is also a non Hamiltonian graph which will satisfy these particular conditions which conditions are necessary conditions which we have stated earlier video lectures, we have proved that this particular Hamiltonian graph has 2 disjoint C5 cycles that is it is having 2 factors of Petersen graph. So, it is not a spanning cycle; that means, it is not single spanning cycle, therefore, it is non Hamiltonian graph although, it satisfies the necessary condition here.

(Refer Slide Time: 21:56)



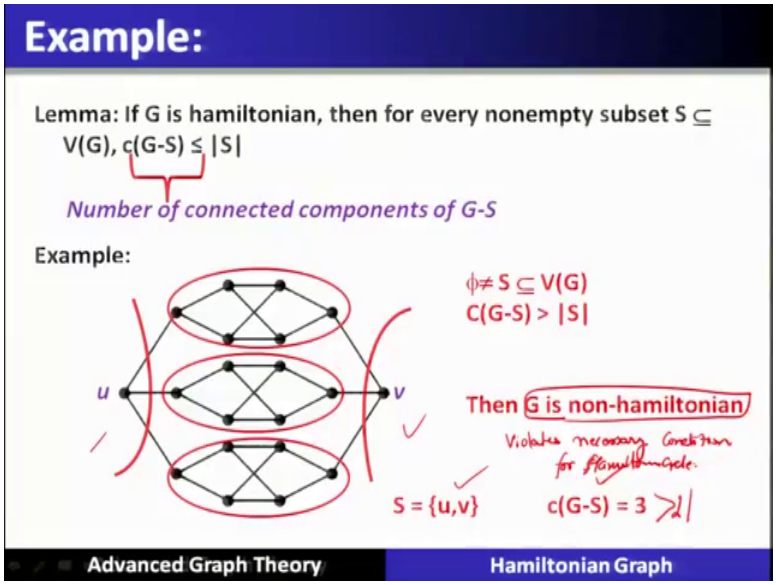
We can see the same necessary condition; that means, if we are given 3 components, this is component number 1, this is component number 2 and this is component number 3.

Let us find out; how many vertices are required for a Hamiltonian cycle to represent. Now as we have seen in the necessary condition that these these are the components and let us make S. So, let us start from this particular vertex and construct a Hamiltonian cycle. So, we start from this particular vertex, visit this particular node, visit this node and this particular node; that means, there is a spanning path which will cover all its vertices and then leave from that vertex, we are leaving this particular component. So, as I have told that when it will leave, it will basically arrive in S on a distinct vertex let us call it as v 1.

Now, from v 1, let us enter again to a different component on a vertex and then visit another node and let us leave this particular component, it will arrive on another distinct vertex, let us call it as v 2, from v 2 we can again enter into another different component, then visit all the vertices on that spanning cycle which we are going to construct and then leave. So, and then leave this particular component. So, when we leave this component it will arrive on S on a particular vertex, let us call it as v 3. Now from v 3, we can join from where we have started and hence we complete the Hamiltonian cycle which requires that v 1 v 2 and v 3 3 different vertices are to be present in the in this particular graph.

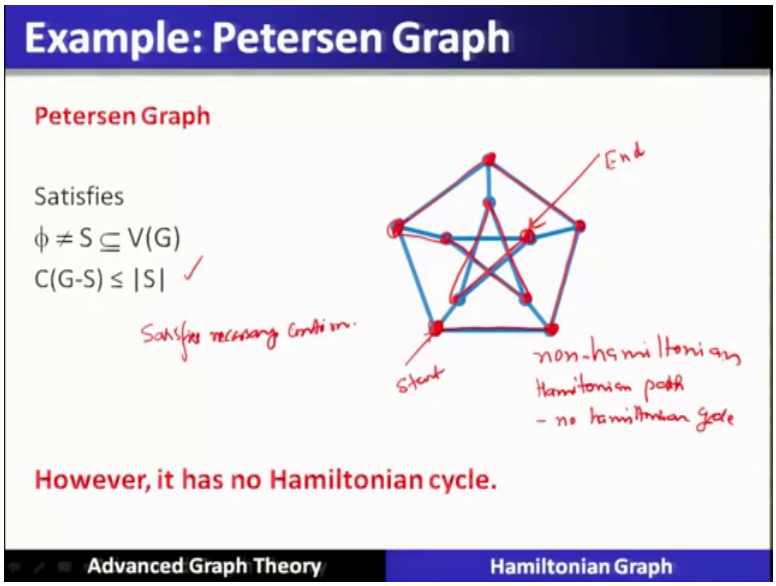
So, this complete graph which we have shown is a Hamiltonian graph and we have also constructed a Hamiltonian cycle. Now to show that it suffice the necessary condition. So, here the size of S is nothing, but 1, 2, 3 and the number of components on is also 1, 2, 3. So, both are equal relation, hence, this is in this particular graph you see that the necessary conditions holds.

(Refer Slide Time: 25:32)



So, again we will show the same example that if we remove these two vertices from this particular bipartite graph then the number of components is 3 which is exceeding the cardinality of S that is nothing but 2. Hence, this is non Hamiltonian why because here it violates the necessity condition for a Hamiltonian cycle to be present hence the graph is non Hamiltonian.

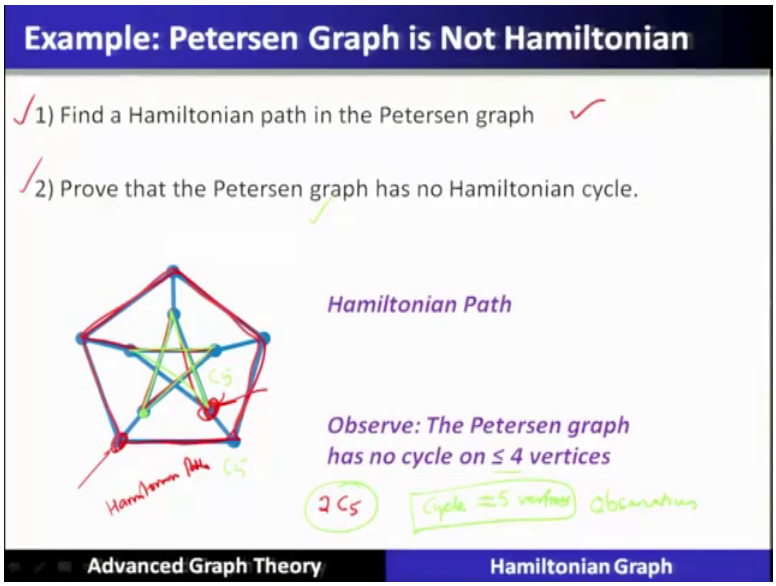
(Refer Slide Time: 26:22)



Let us take the Petersen graph Petersen graph also satisfies this particular condition which is nothing, but the lemma for the Hamiltonian cycle which is necessary condition is satisfied; that means, for a any subset of vertices plug out the number of components will be basically the size of will be the at most the size of that set S.

Hence, it satisfies the necessity condition. So, we will show here that the necessity conditions is not sufficient for a for a Hamiltonian cycle to be present in a graph. Let us see here in this particular example that let us see that we start at this particular vertex, then we visit the vertices around the the cycle C5 which is present. Now, we enter into the inner cycle through this particular vertex. Now here we will end. So, this particular contraction will give you the Hamiltonian path, but we cannot complete the circle or a circuit. Hence, there is no Hamiltonian cycle and therefore, Petersen graph is non Hamiltonian.

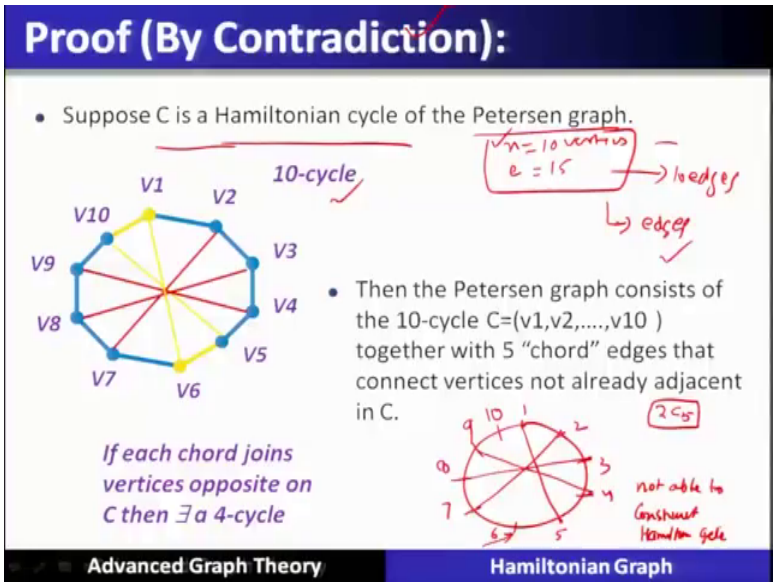
(Refer Slide Time: 28:14)



We can also shown that it has the Hamiltonian path; that means, we can start from this vertex we can trace back all the vertices enter into it then visit and finish it over here, this is the Hamiltonian path that we have done second part is basically that we have to prove that Petersen has no Hamiltonian cycle. Now, we have earlier seen that the Petersen graph in the previous videos that Petersen graph has 2C5; that means, 2 disjoint cycles of C5 that is one cycle is outer cycle the other cycle is the inner cycle. So, this is C5 cycle to C5 cycles are there in Petersen graph there are two C5 cycles are there.

So, the cycle of size 5 of 5 vertices present in the Petersen graph; that means, it does not have the cycles less than 5 that is 4 or less than that cycles are not present in the Petersen's graph. So, this is the observation number 1.

(Refer Slide Time: 29:48)



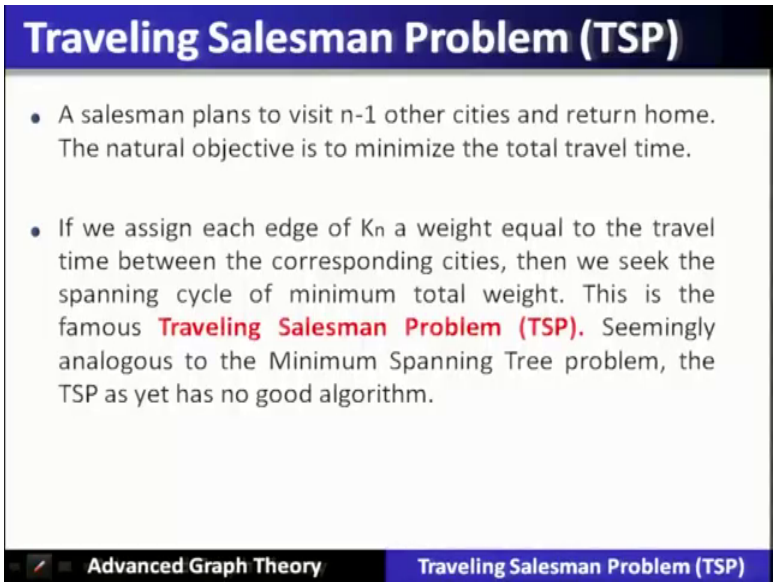
Let us use this observation to prove that Petersen's graph has no Hamiltonian cycle Petersen graph has number of vertices has 10 different vertices and it has number of edges as 15 different edges are there.

So, 15 edges are there and 10 vertices are there, let us construct Hamiltonian cycle. So, we will prove by contradiction, let us say that C is a Hamiltonian cycle Hamiltonian cycle is a cycle which is a spanning cycle of that particular graph; that means, it will span through all the 10 vertices, let us form a 10 cycle and now we will place an arc. So, this is the 10 cycle. Now you know that there are 2C5 cycles are present on this Petersen graph. So, 1 and 5 can have a chord. So, 10 different edges, we have used in the cycle edges, we have used in the cycle remaining 5 edges, we have to put as a chord. So, this is the first possible chord.

Now, there is no 4 cycle or less than 4 cycle. So, hence these chords are not possible. So, this is one possible chord. Now let us take 2, let us take 3, let us take 4, now the last one cannot be basically included why because there is there cannot be cycle of length 4 or less than that.

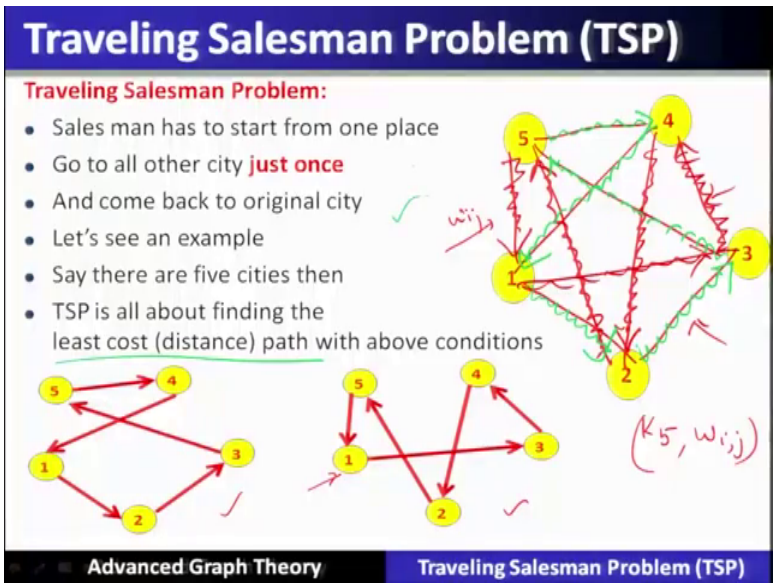
Hence, we are not able to construct a Hamiltonian cycle here in this particular graph.

(Refer Slide Time: 32:25)



Travelling salesman problem travelling salesman problem deals with a salesman who planes to visit n minus one other cities and return home the natural objective is to minimize the total travel time. Now, if you consider a graph that is a complete graph Kn and we assign each edge with a weight equal to the travel time between the corresponding cities, then we seeks the spanning cycle of minimum total weight. This is the famous travelling salesman problem seemingly analogous to the minimum spanning problem, but the travelling salesman problem as at as not good algorithm.

(Refer Slide Time: 33:22)

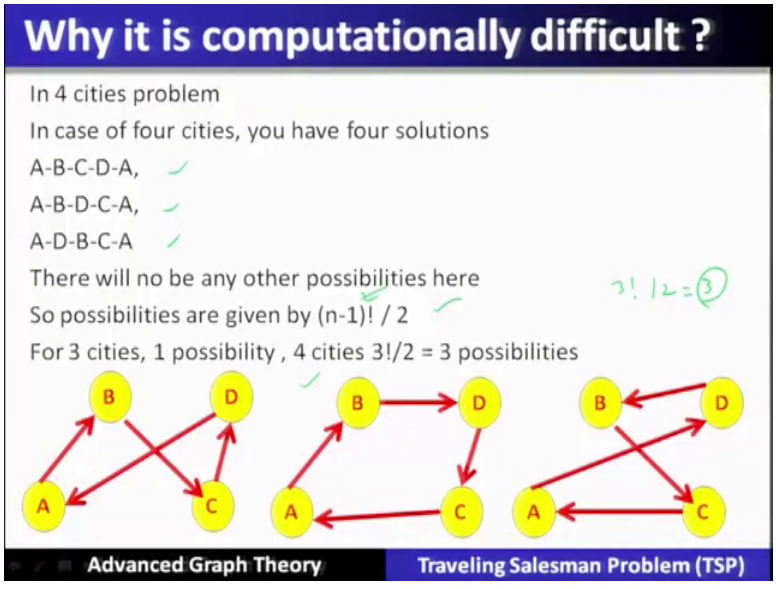


Let us take understand travelling salesman problem through a simple example, here we will consider 5 different nodes, we can draw a complete graph on 5 different nodes. so, let us say this is K 5. Now here, we will find out and we will assign the weights, let us say W i j is assigned to all these i js the weights. So, the salesman has to start from 1 point, go to all other cities is exactly once and come back to the original city here, these are the possible traces, for example, start from 1, go to 3, choose by this edge, then go to 4 by this edge, then from 4, it will go to 2 by this edge and from 2 to 5 by this edge and from 5 to 1 back from where it is started, this is one such possibility another way is that it will start from 1

Taking another from 1, it will go to to 2 and from 2, it will go to 3, from 3, it will go to go to 5, from 5, it will go to 4, from 4, it will come back to 1 from where it has started.

So, travelling salesman is about finding the least cost path. So, that all the cities are visited at exactly once and it will return back to the same points.

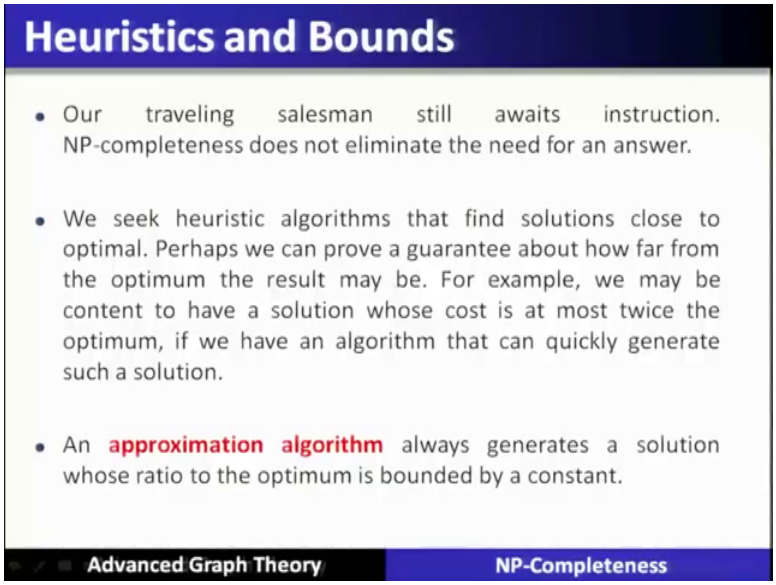
(Refer Slide Time: 35:45)



Now, here we we will see that there are various possibilities exists there is more than one ways are travelling, salesman can visit the cities and can come back of all these different alternatives possibilities, it has to find out the one with a least cost. So, let us see the how many possibilities, it has to explore that is n minus 1 factorial by 2, even for the 4 cities, it comes out to be 3 factorial by 2 that is by 3 possibilities as we increase the number of

nodes in the problem, this particular search space becomes exponential in size and becomes a typical problem cannot be easily solved in this manner.

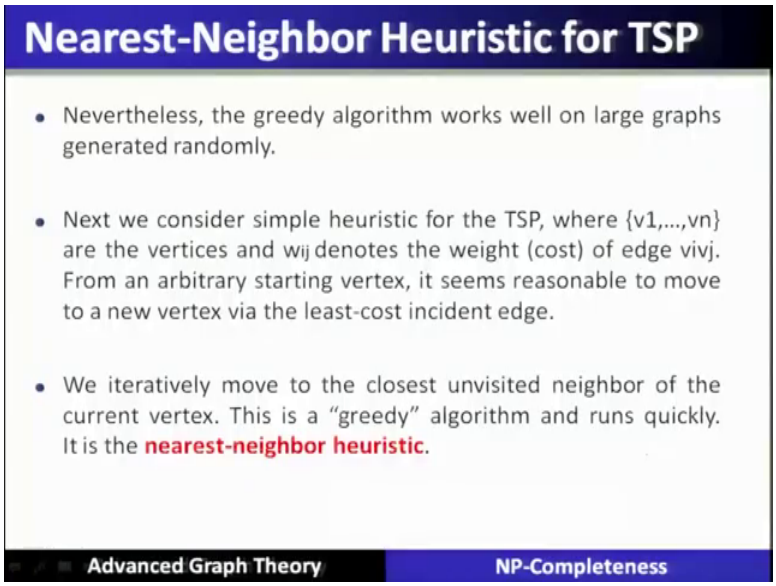
(Refer Slide Time: 36:31)



Therefore different heuristics and bounds are given to solve the the traveling salesman problem. So, our traveling salesman is still awaits the instructions that is NP-Completeness does not eliminate the need for an answer because it is a practical application and many applications are based on this particular solution of this particular problem. So, we seek a heuristic algorithm that finds the solution close to the optimal why because finding a optimal may take a too much of time perhaps we can prove a guarantee about how far from the optimum the result may be, for example, we may we may be the content to have a solution whose cost is at most twice the optimum if we have an algorithm that can quickly generate a solution.

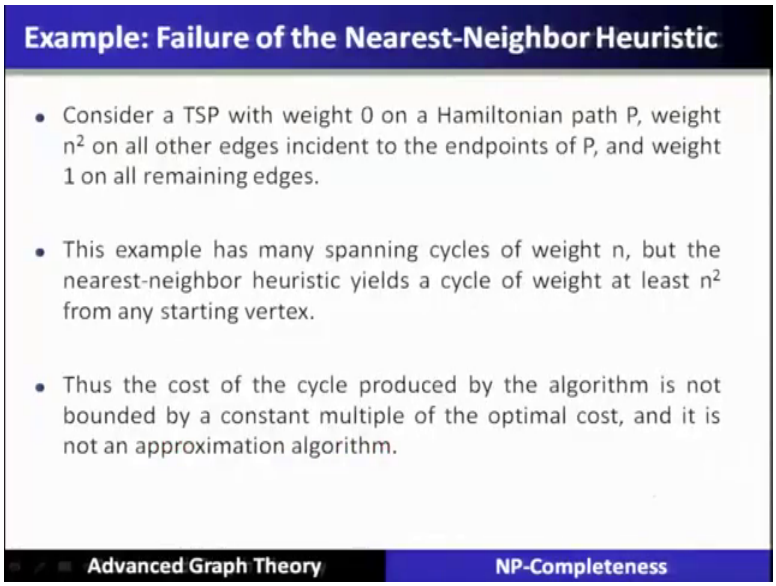
So, such possibilities; that means, deviation from the optimum, but with their particular guarantees are also a practical interest, therefore, deviation from the optimum with a particular guarantees those algorithms which compute travelling salesman problem efficiently are called approximation algorithm the approximation algorithm is always generates a solution whose ratio to the optimum is bounded by a constant value, if it is not then those algorithms are not approximation algorithm.

(Refer Slide Time: 38:08)



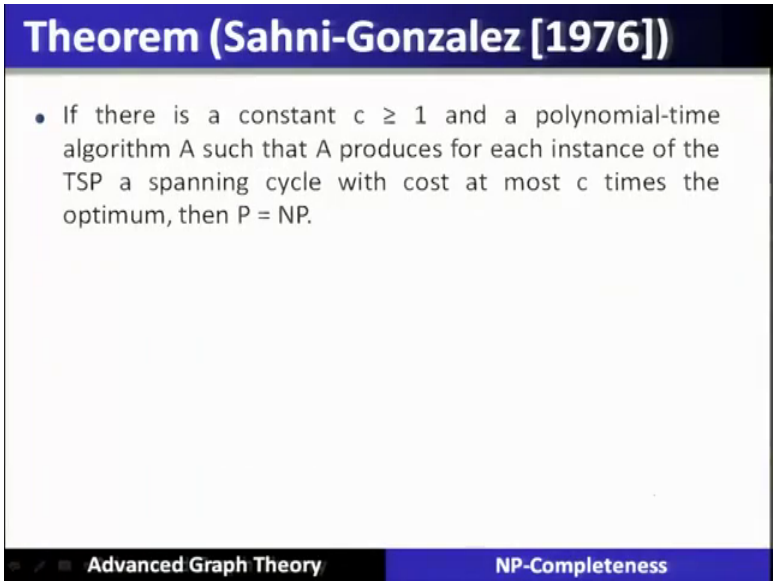
Let us take one such heuristics that is called nearest neighbor heuristic for travelling salesman problem nevertheless the greedy algorithm works, well on large graph generator randomly. Next, we consider a simple heuristic for travelling salesman problem where the vertices v 1 to v n are the vertices and W i js denote the weights via the cost of the edges between vertex v i and vj. Now from an arbitrary starting point, it is seems reasonable to move to the next vertex, why are the least cost incidence edge, therefore, we iteratively move to the closest unvisited neighbor of the current vertex and this is the greedy algorithm and run quickly and this is called the nearest neighbor heuristic.

(Refer Slide Time: 39:05)



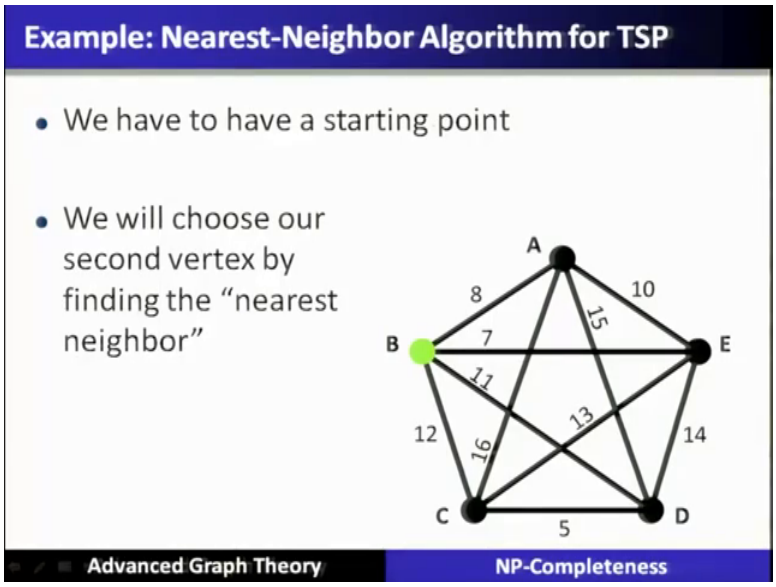
Consider a travelling salesman problem with a weight 0 on the Hamiltonian path weight n square on all other edges incident to the end point of P and weight 1 on all the remaining edges, this example has many spanning cycles of weight n, but the nearest neighbor heuristic yields a cycle of weight at least n square from any starting point. Thus, the cost of the cycle produced by the algorithm is not bounded by the constant multiple of optimum optimal cost and it is not an approximation algorithm.

(Refer Slide Time: 39:44)



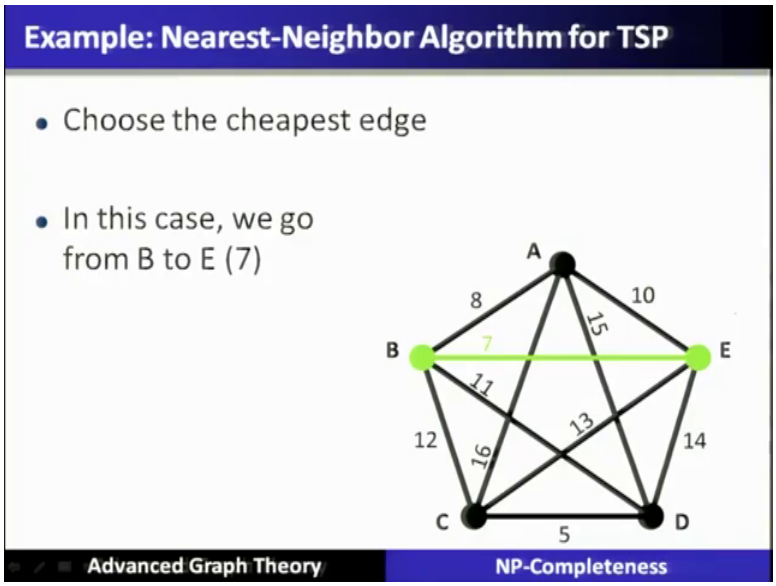
So, there is a theorem which says that if there is a constant C which is at least one and a polynomial time algorithm a such that a produces for each instance of a travelling salesman problem a spanning cycle with a cost at most C times the optimum value, then P is equal to NP; that means, this particular problem also becomes difficult to be computed with the approximation algorithm. Let us take the example of a nearest neighbor algorithm for travelling salesman problem.

(Refer Slide Time: 40:21)



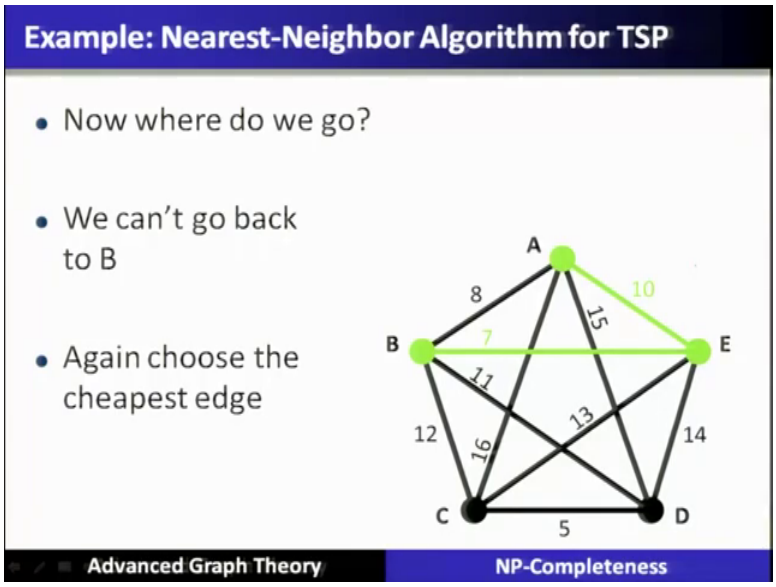
Now, we have to have a starting point. Let us say that the starting point is shown in a green color vertex as B. Now, we will choose our second vertex by finding the nearest neighbor. So, here the nearest neighbor means the neighbor with a minimum value or minimum cost here, there are possibilities 1, 2, 3, 4 and among them, this is having the minimum cost.

(Refer Slide Time: 41:02)



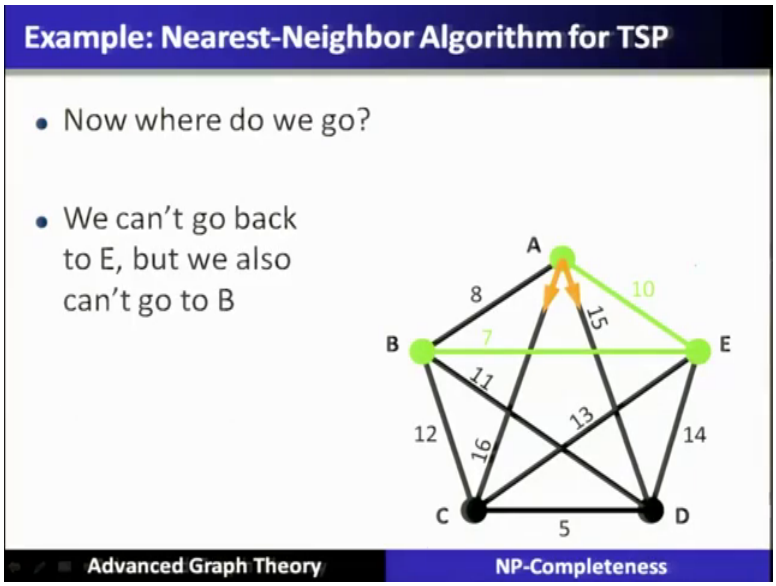
So, we will place an edge from B to E. Now after reaching E, now we have the possibilities we can go to A, we can go to C, we can go to D. Now among them, this C is basically with with the minimum or the least cost; that means, from E.

(Refer Slide Time: 41:23)

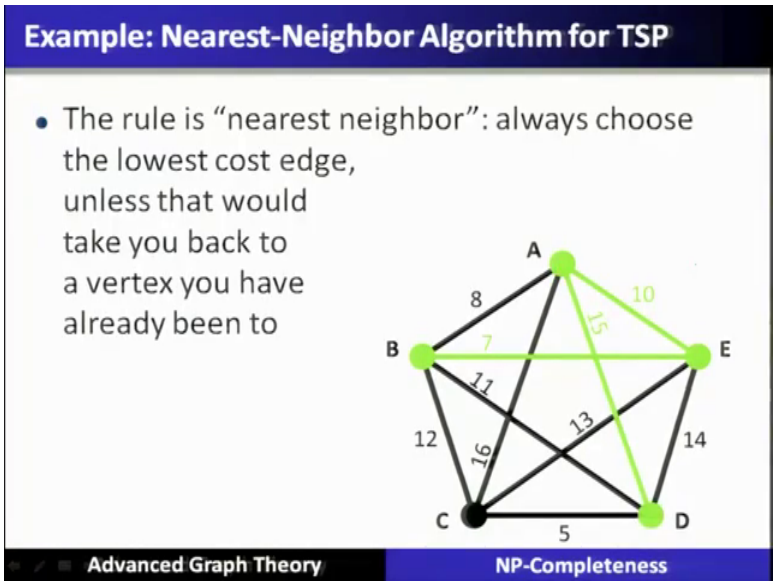


We can go to A with a minimum cost of 10. Now after reaching A, we can choose between C and D. So, we will choose D, why because it is having the minimum cost after reaching D, we have the possibilities to choose.

(Refer Slide Time: 41:31)

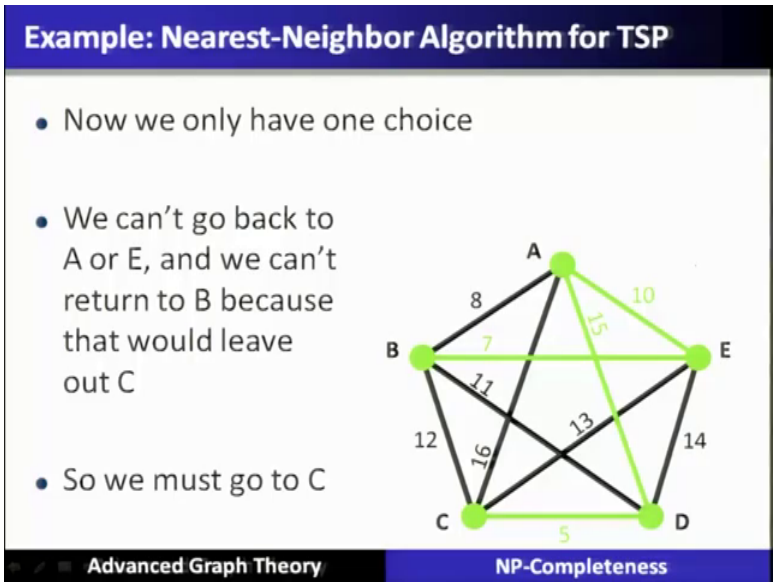


(Refer Slide Time: 41:35)



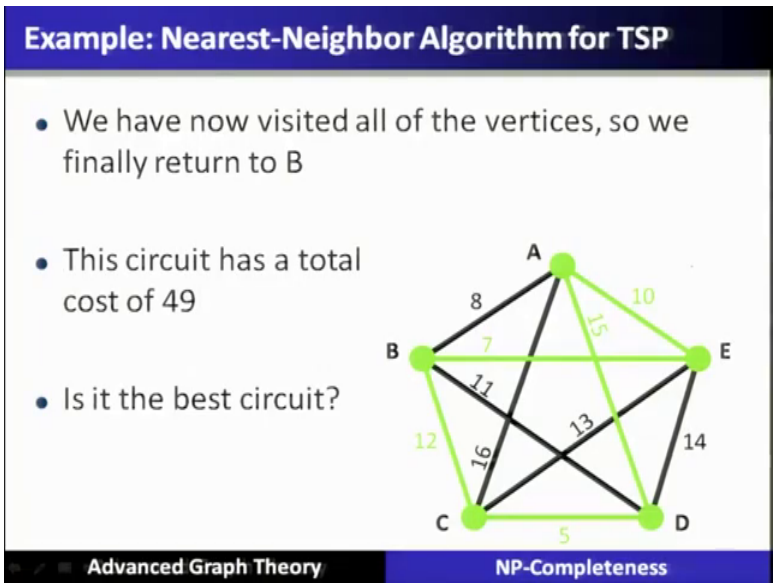
So, C is not wasted. So, we will choose C and from C, we have go to B again.

(Refer Slide Time: 41:39)

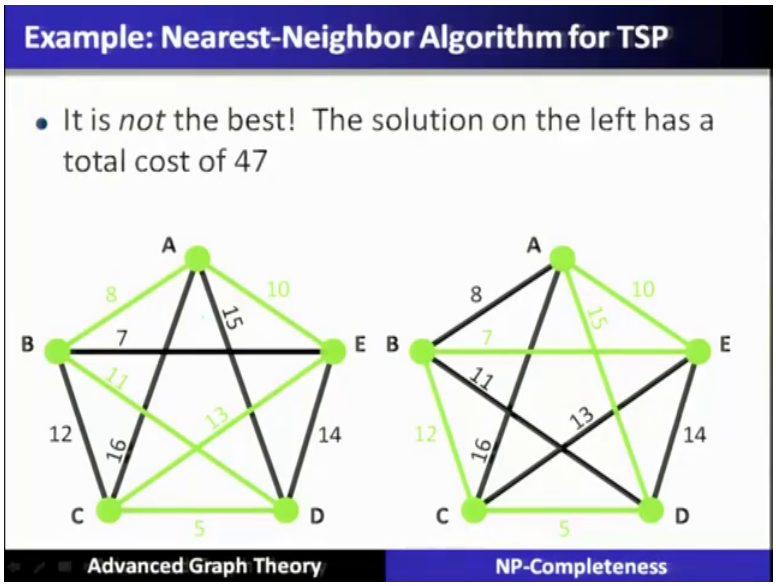


So, the total cost, if we sum these green color edges, the weights on this green color edges that total cost comes out to be 49.

(Refer Slide Time: 41:54)

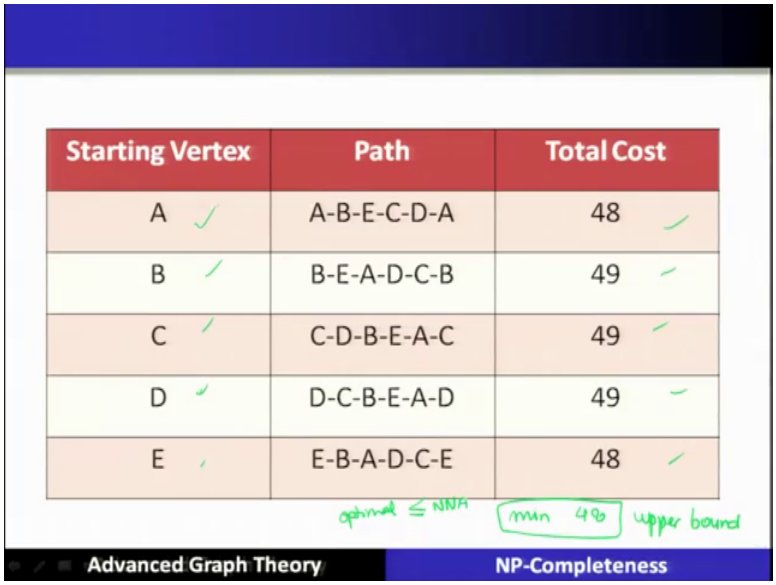


(Refer Slide Time: 42:04)



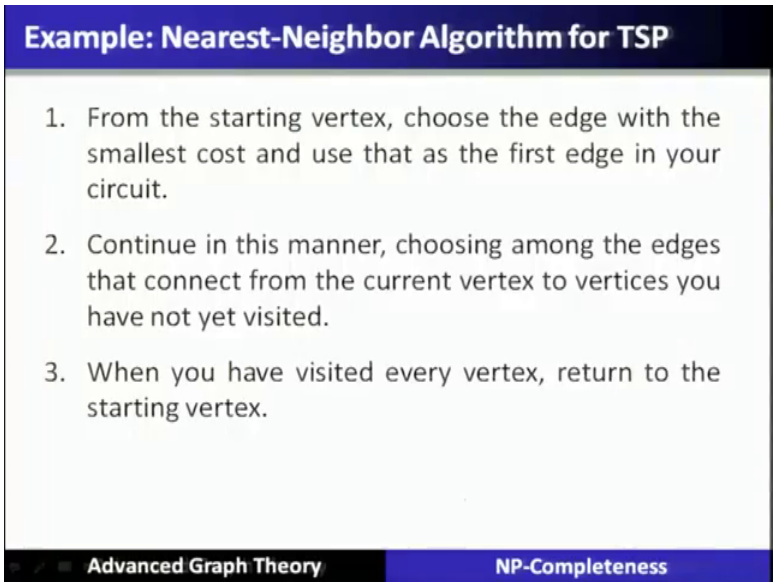
If we choose another way; that means, another starting point, then the cost can also come down to 47.

(Refer Slide Time: 42:14)



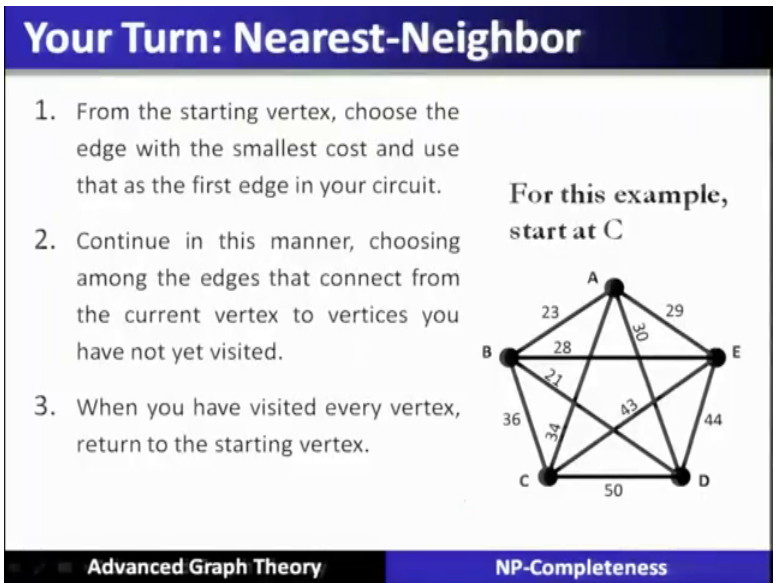
So, here in this case, if we choose different starting vertices and use the nearest neighbor heuristics, the total cost we can see that comes out to be 48, 49, 49, 48. So, minimum is 48. So, by nearest neighbor heuristics the minimum cost which we have obtained it called an upper bound solution so; that means, the optimal solution is upper bounded by this nearest neighbor algorithm that we have obtained in this particular way.

(Refer Slide Time: 43:01)



So, let us review this nearest neighbor algorithm from a starting vertex. Choose an edge with the smallest cost and use that as the first edge in the circuit, continue in this manner choosing among the edges that connects from the current vertex to the vertices that we have not yet visited. So, when you have visited every vertex, return to the original or to the initial starting vertex.

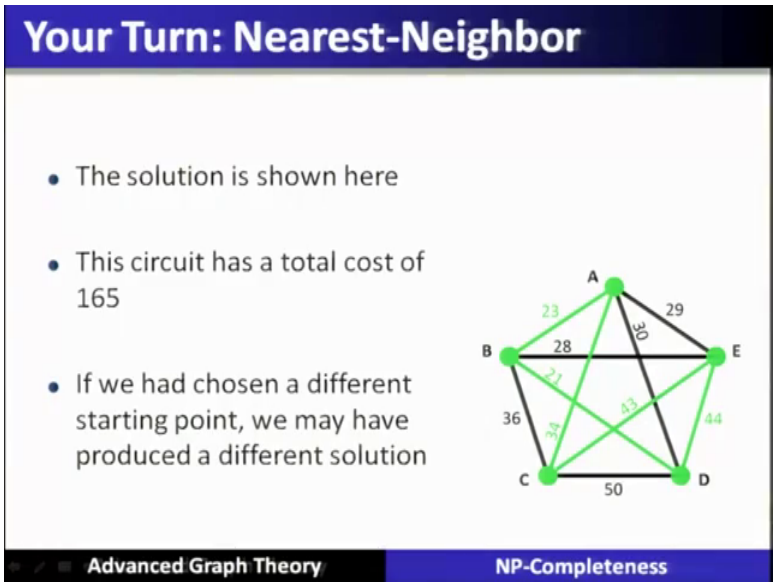
(Refer Slide Time: 43:30)



So, this particular another example, we can see here in this particular example, we start from C, we have 34, 43, 50. So, we will go to A, when we reach A, we have 23, 30 and

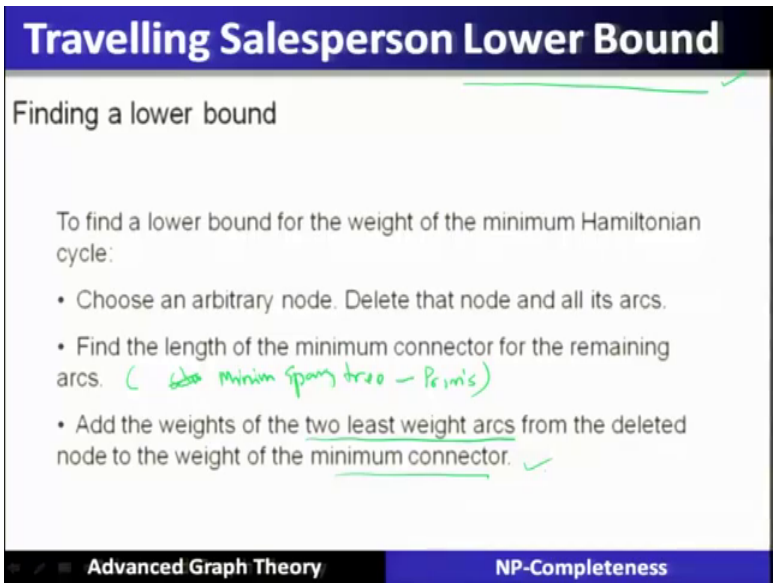
29, we will go to B, from B, there is a possibility of 28, 21. So, we will choose 21 and then we will reach D, from D, we have a possibility to go to E. So, we will go to E, from E, we have to come back to C. Now, we have to see how to come back to C, there is a edge of 43. So, we will come back to C here in this particular case.

(Refer Slide Time: 44:26)



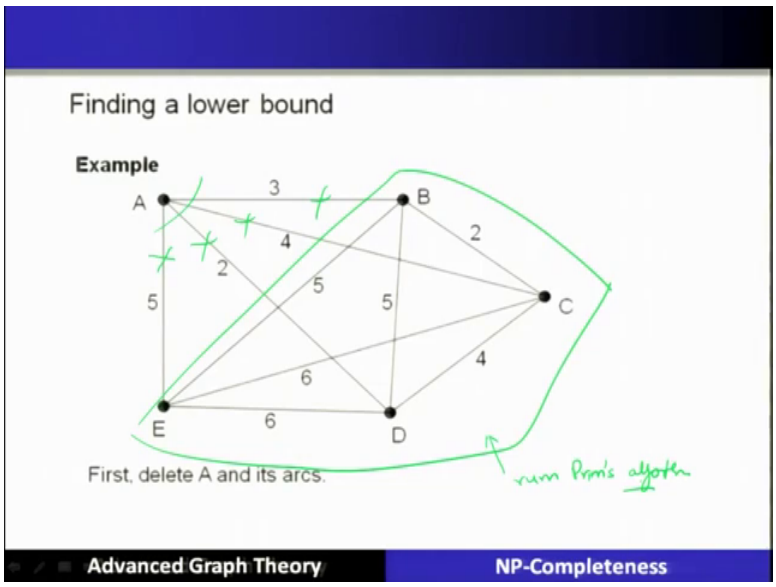
So, all these things; the solution is shown over here where the total cost comes out to be 165 of this particular circuit which we have identified trough travelling salesman problem.

(Refer Slide Time: 44:37)



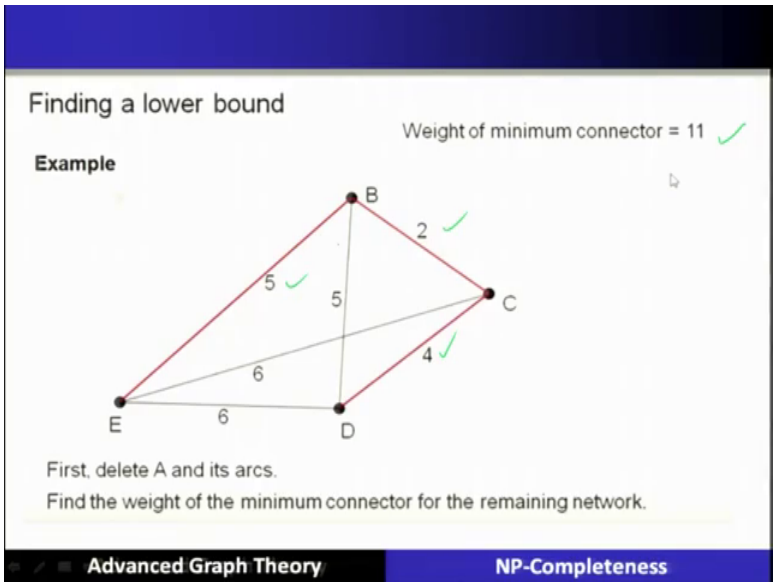
Now, let us find out the lower bound upper bound we have already computed through a nearest neighbor algorithm. So, to find a lower bound for the weight of the minimum Hamiltonian cycle, we choose an arbitrary node and delete that node and all its arcs then find the length of the minimum connectors for remaining arcs and we will use any minimum spanning tree algorithm or a either Prin's algorithm, we will use it and once the tree is constructed, then we will place the two arcs with a least weights from the deleted nodes to the minimum spanning tree which we have obtained and that will be the lower bound.

(Refer Slide Time: 45:32)



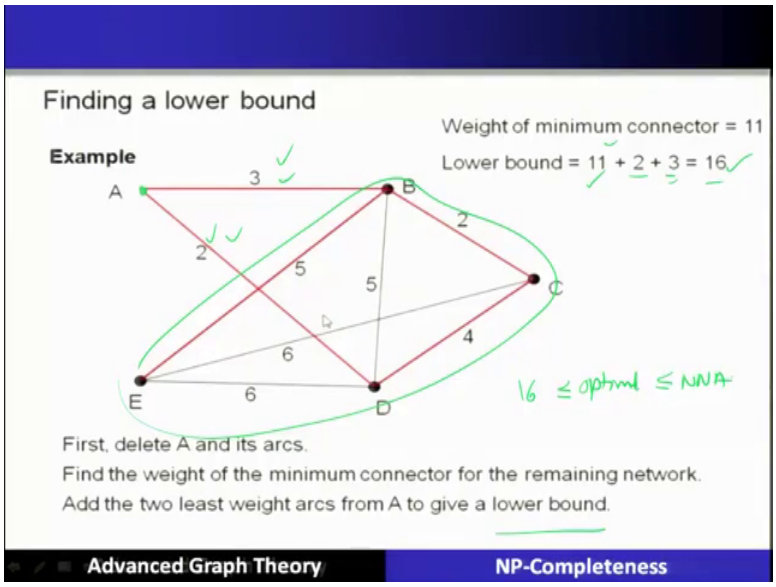
Let us take this particular example to understand the lower bound competition for a travelling salesman problem on a given graph with a given weights. Now you see that this particular graph is a complete graph of 5 vertices. So, here we are going to delete A and all its arcs will be deleted. So, the remaining part of the graph after deleting a will be this much and now, here we have to run the Prin's algorithm starting point we have to choose the starting point.

(Refer Slide Time: 46:23)



And let us say that we will construct a minimum spanning tree which is shown in the red color which will span through all the vertices. So, if we find out the total cost total cost is 4, 2 and 5, if we add; it comes out to be 11.

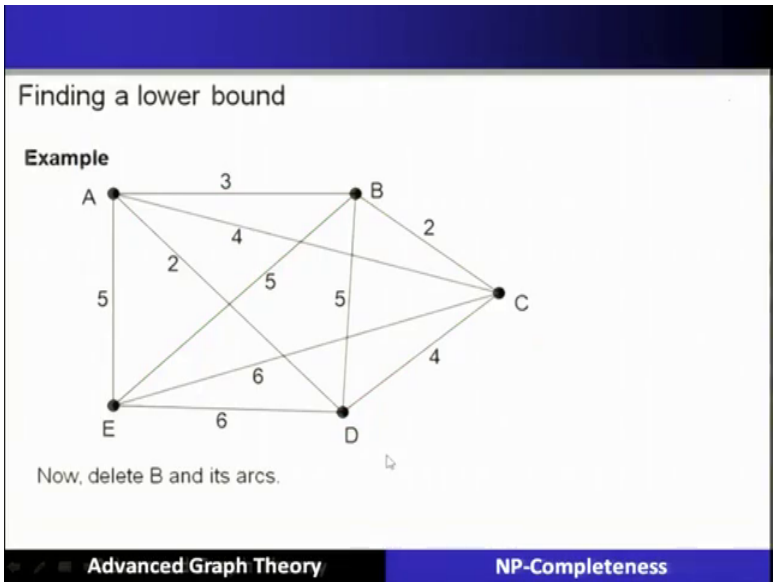
(Refer Slide Time: 46:48)



Now, we will place the two edges. So, it will be 11 now as for as A is concerned, we will place two edges with a least two different cost; that means, AE was having higher cost. So, it is not included. So, two edges are added up 2 and 3, similar cost of the lower bound is 11 of this particular competition plus 2 more edges, we have added 2 and 3 that

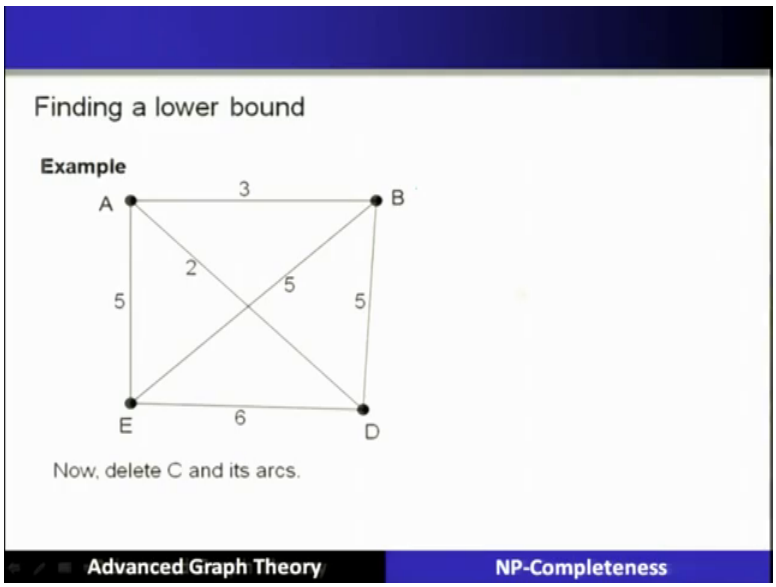
comes out to be 16. So, the lower bound, we have obtained is 16. So, the optimum or optimal solution should be at least 16 and it will be upper bounded by that particular value which we have obtained using nearest neighbor algorithm.

(Refer Slide Time: 47:34)

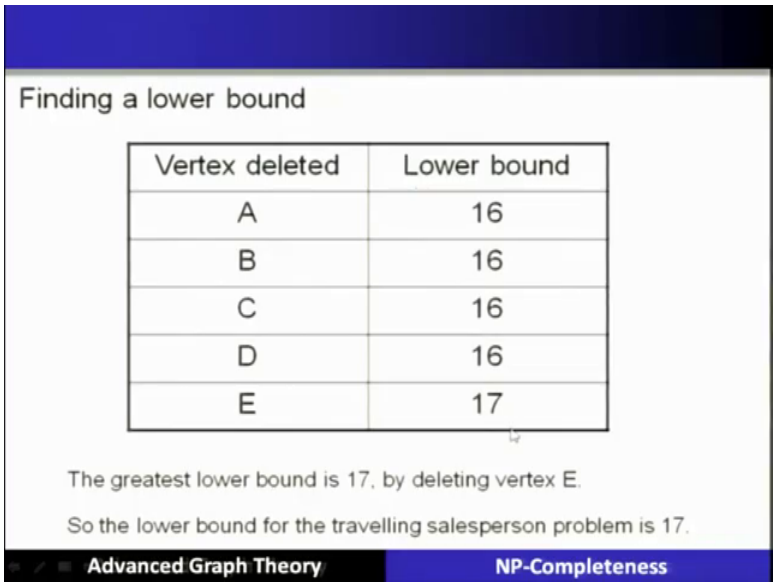


Now, we can delete some other vertex, let us say A B and again we run this algorithm, we obtain a different value here also, we obtained 11 6.

(Refer Slide Time: 47:46)

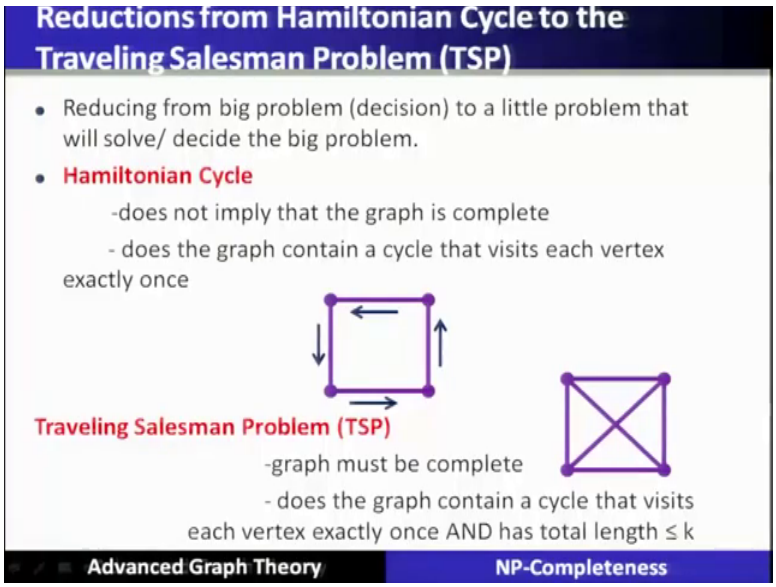


(Refer Slide Time: 47:50)



And now we delete C and now, if you see the statistics of competition of a lower bound how to compute the lower bound is one vertex a is deleted the value comes out to be 16 when B is deleted comes out to be 17 C is deleted 16 D is deleted 16, when E is deleted it comes out be. So, the greatest value among all is 17. Hence, the lower bound becomes 17.

(Refer Slide Time: 48:30)

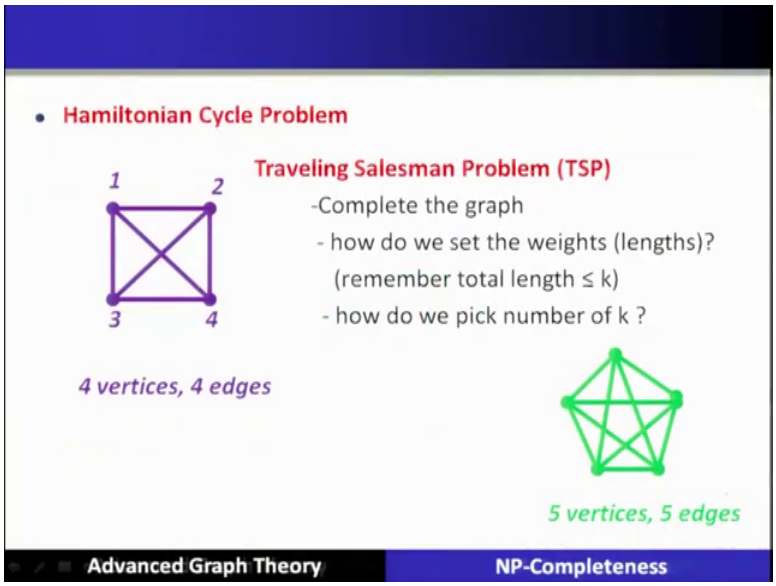


And then we will apply this one optimal algorithm on this values reduction from Hamiltonian cycle to the travelling salesman problem. So, reducing from the big problem to a little problem that will solve a big decide the bigger problem.

So, let us see this particular motion for reducing the Hamiltonian cycle to a travelling salesman problem. Now Hamiltonian cycle problem setting does not require the graph of to be complete does the graph contains the cycle that visits each vertex exactly once; that is only requirement of the problem setting to find out the Hamiltonian cycle. Let us take this particular graph to find out Hamiltonian cycle in the graph, it need not be complete graph, we start from a point visit to another node visit to next node and finally, come back to the same point this is the cycle which is is a spanning cycle, we require that is called Hamiltonian cycle which we are searching on this particular problem setting about the problem setting in a travelling salesman, the graph must be complete; that means, the missing edges which is not there in Hamiltonian are to be present to make the graph complete.

Second thing is that it does the graph contains the cycle that visits each vertex exactly once and also should have the cost of total tour is basically less than or equal to k for some given value of k so; that means, it is not any such cycle, but a cycle which is having the total cost less than K. Those kind of cycles are being required here in travelling salesman problem settings.

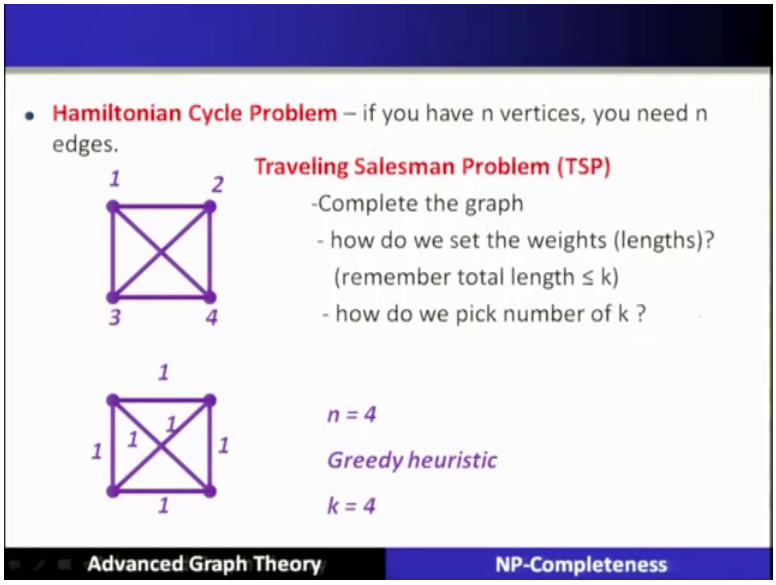
(Refer Slide Time: 50:30)



So, with this particular difference; so, we are given the Hamiltonian cycle problem, then we can convert it into a problem formulation in the travelling salesman in the following manner. We have to first complete the graph in the sense the missing edges we have to add. So, that the graph becomes a complete graph. Now the question is if we add the edges, then we require some weights also to be to be there.

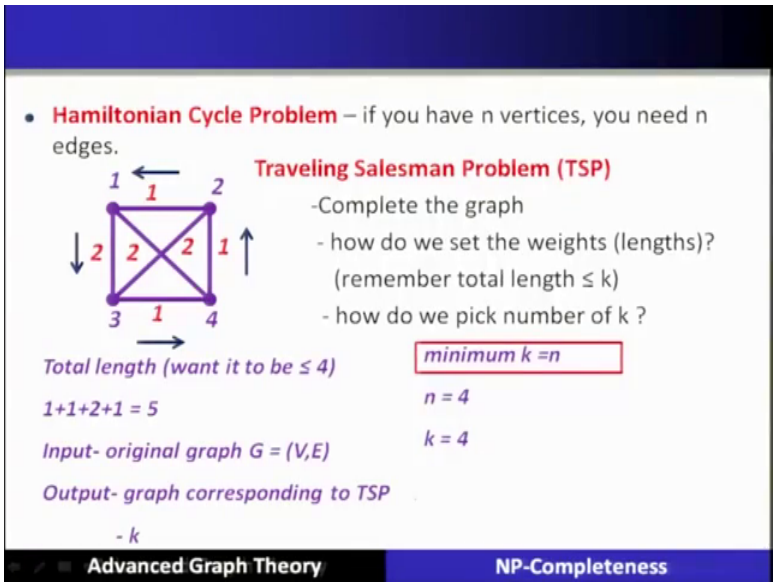
So, we have to about this particular weight and also we have to choose a number because the total cost of the tour also is to be bounded by some number k and that number k is also required to be mentioned in the travelling salesman problem solution.

(Refer Slide Time: 51:22)



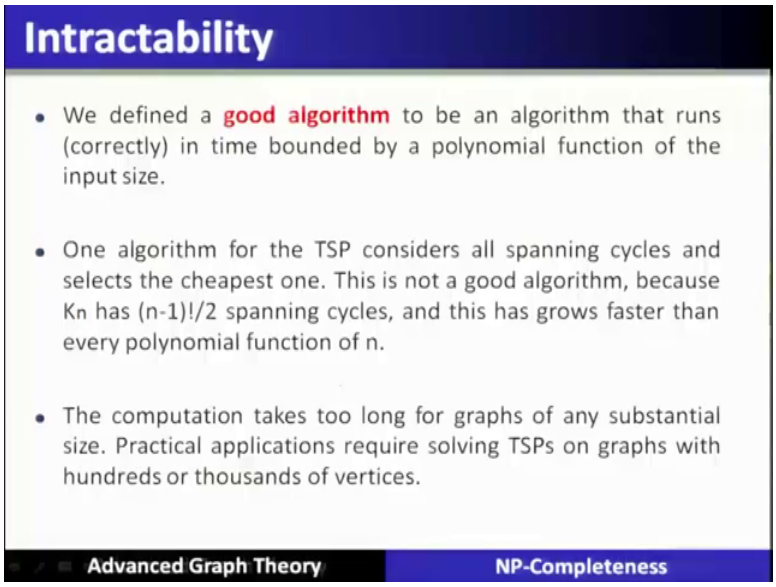
Now, the value of k, for example, if this particular graph is given and all the edges are having edge weights has one, then if we find a cycle in that in the this one travelling salesman problem, then this cycle is of length or a cost 4. So, if we say that find out a cycle of the cost less than 4 that is not possible. So, we have to choose the number for k such that we can find out existing cycle with a minimum cost.

(Refer Slide Time: 52:06)



Now, if you change the weights of the edges and we mention that we want a cycle to be of length four. So, the what what should be the minimum value of k to be there in this particular graph, we have to certain that value of k NP completeness.

(Refer Slide Time: 52:34)

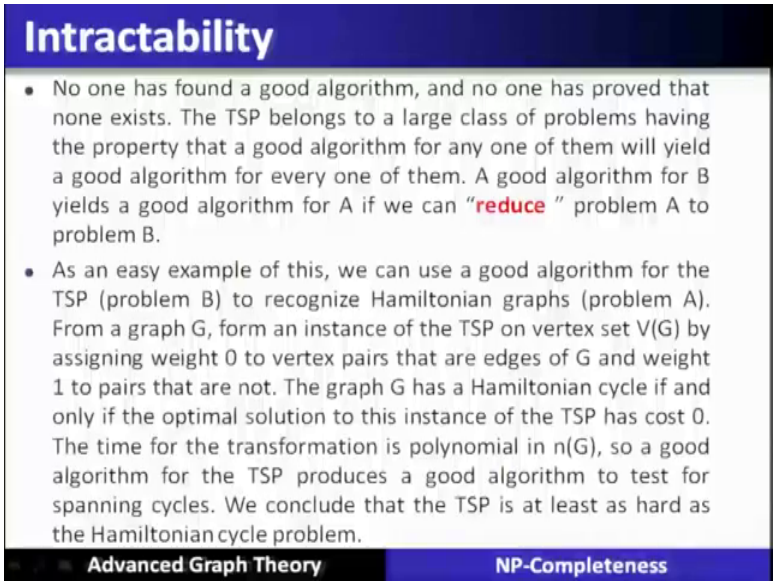


Intractability; we defined a good algorithm to be the algorithm that runs correctly; obviously, in time bounded by the polynomial function of the input size those are called good algorithms now one algorithm for travelling salesman considers all the spanning cycles and selects the cheapest one that is called a brute force method will not be a good

algorithm because the complete graph of an node Kn as n minus one factorial by two different spanning cycles. So, for a given input size n the total time which will require is not a polynomial function, but it is an exponential function why because here n minus one factorial by 2 and this has growth faster than every polynomial function n.

So, this algorithm is not a good algorithm which we are looking for hence the competition takes too long for the graphs of any substantial size practical applications require solving travelling salesman on a graph with hundreds of thousands of a vertices where this particular value shoots up because it is not a polynomial. Hence, it becomes impractical to solve this particular problem using this kind of algorithm. So, are we searching for a good algorithm?

(Refer Slide Time: 54:15)



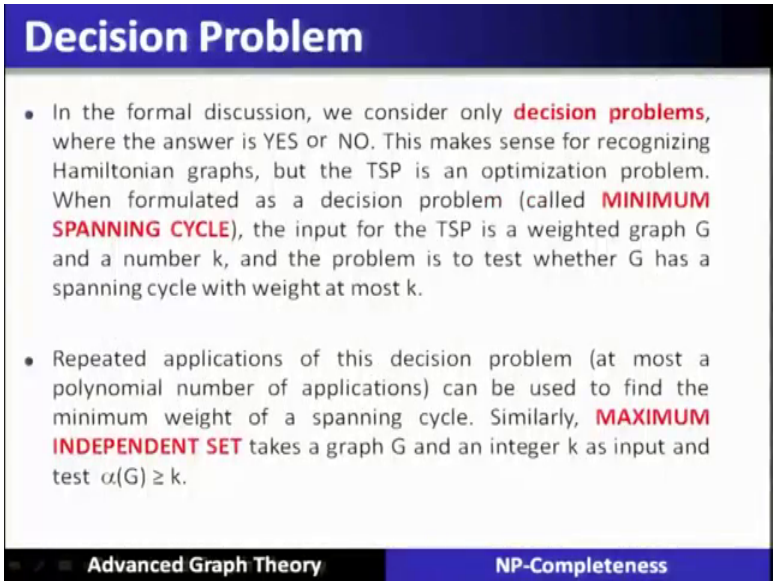
So, no one has found a good algorithm for travelling salesman problem and no one has proved that none of this algorithm exists the travelling salesman belongs to a large class of problems having a property that having a good algorithm for any one of them will yield the good algorithm for every one of other such problems for which the good algorithms are not yet known.

So, a good algorithm for B will yield a good algorithm for a if we can reduce the problem A to problem B as an easy example of this we can use a good algorithm for a travelling salesman problem; problem B, for example, it is called to recognize the Hamiltonian graphs that is problem A. Now from the graph G form an instance of

travelling salesman problem on a vertex at G by assigning the weight 0 to the vertex pairs that are the edges of G and weight 1 to the pairs that are not present. So, the graph G has a Hamiltonian cycle if and only if the optimum solution to this instance of travelling salesman problem has a cost 0.

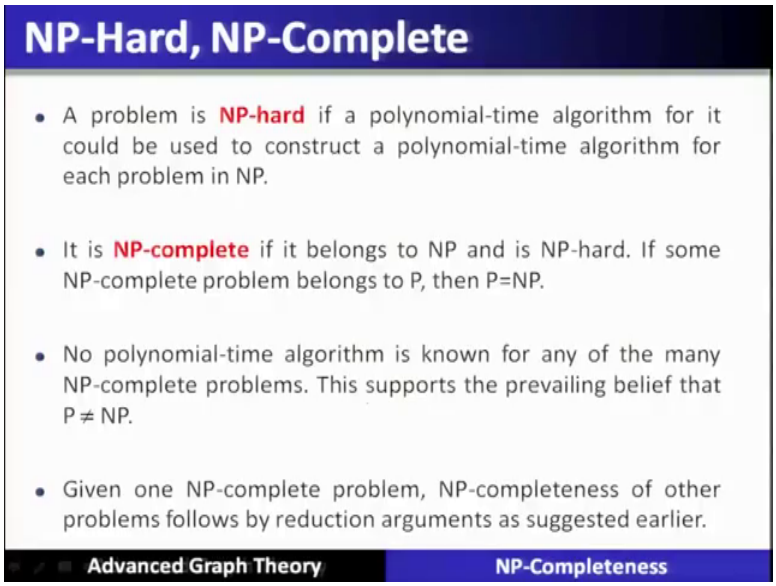
So, the time for transformation is polynomial in n of G. So, a good algorithm travelling salesman produce a good algorithm for to test for the spanning cycles, we conclude that travelling salesman is at least as hard as the Hamiltonian cycle problem.

(Refer Slide Time: 55:44)



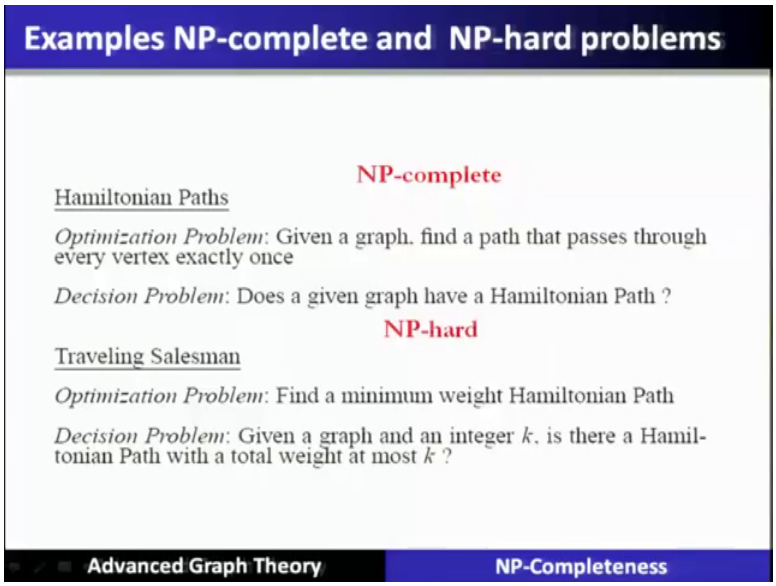
So, in the formal discussion, we consider only the decision problem where the answer is yes or no, this make sense for recognizing the Hamiltonian graphs, but travelling salesman is an optimization problem; that means, finding a tour with a with a minimum cost. So, when formulated as a decision problem called minimum spanning tree the input for that travelling salesman is a weighted graph G and a number k and the problem is to test whether the graph as a spanning cycle with a weight at most k repeated application of this decision problem at most a polynomial number of applications can be used to find the minimum weight of a spanning tree. Similarly, the maximum independent set takes a graph and an input k as the input and test alpha G is at least k.

(Refer Slide Time: 56:37)



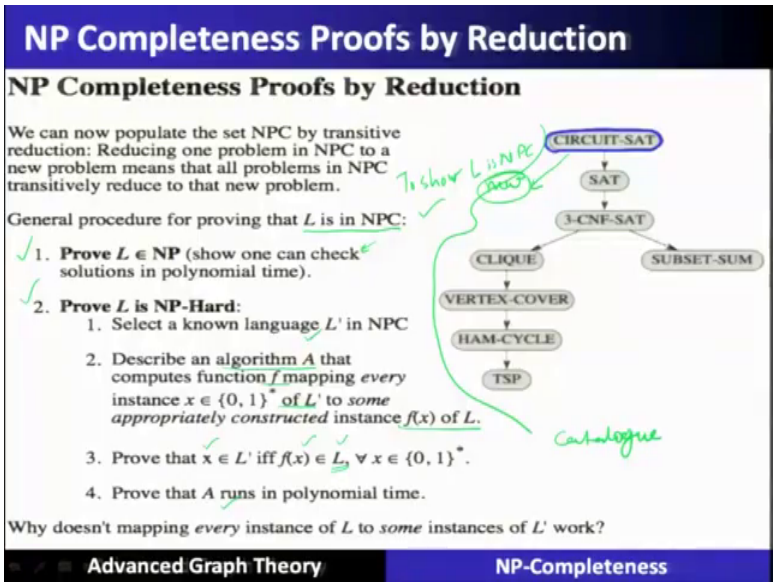
So, those problems are called intractable problem why because the good algorithms are not known and if one knows a good algorithm of any of these particular problem then the other problem can be reduced and solved hence we are now going to give the farther definition in this class of problems. So, NP hard and NP complete the problem is NP hard is the polynomial time algorithm for it could be used to construct a polynomial time algorithm for each problem in NP so; that means, those problems are hard as other problems and that is why this problem is called as NP hard. So, it is NP complete if it belongs to NP and it is also NP hard if NP complete problem belongs to P, then P is equal to NP no polynomial time algorithm is known for any of the many NP complete problem this suggests the prevailing belief that P is not equal to NP.

(Refer Slide Time: 57:48)



So, given one NP complete problem NP-Completeness of the other problems followed by the reduction arguments as suggested earlier, we will use if in further discussions. Now examples NP complete NP hard problems, we will see the Hamiltonian path problem the optimization, it is a optimization problem that is given a graph find a path that passes through every vertex exactly once the decision version is that does a graph contains a Hamiltonian path in a f know problem. This is NP complete another example which is called an NP hard problem that is called a travelling salesman problem. Here the optimization version of this problem is to find out the minimum weight Hamiltonian minimum weight travelling salesman problem or minimum weight Hamiltonian path, the decision version of the travelling salesman problem says that given a graph and a integer k is there Hamiltonian path with a cost at most k that is to give NP hard.

(Refer Slide Time: 58:49)



Now, let us go in more details about the unknown problems, how we are going to prove to be NP complete by the method of reduction that is called polynomial time reduction. So, we can now populate the set of NP complete problem by transitive reduction. So, reducing one problem in NP complete problem to a new problem means that all the problems in NP complete class transitively reduce to that particular problem. So, the general problem for proving that a new problem L is in NP complete class or not the generalist app is to prove that for that the two steps are required to prove to show whether L is NP complete problem.

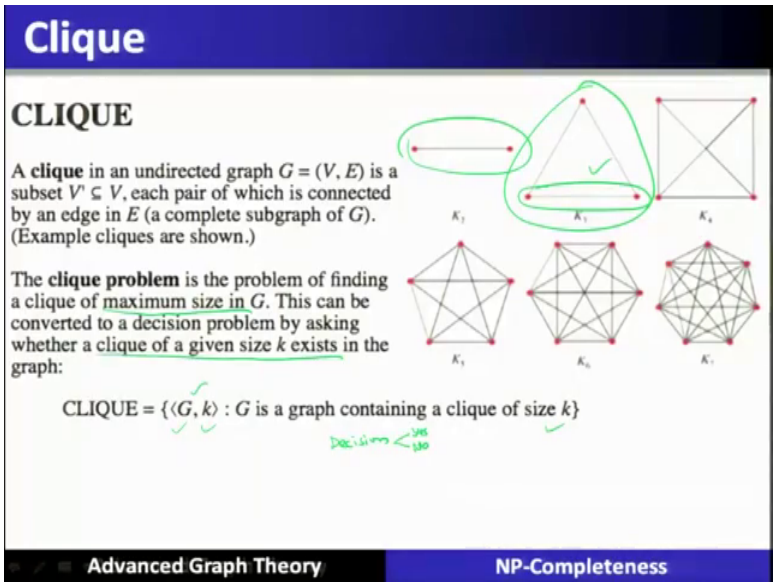
So, there are two steps first step is to show that the problem L is an NP. So, this becomes easy why because of we then get a certificate, then we can check this certificate in a polynomial time whether it is belongs to that particular problem or not solution of a problem or not that we can check in. So, the problem we have to show that is NP; we have to check that solution in polynomial time, then the problem is in NP; second thing is second step for proving NP-Completeness is that we have to prove that the problem value is NP hard for that we have to select a known problem.

So, we require a catalog of different NP complete problems; that means, we know some known NP complete problem and we select one such problem. Let us say L prime. Now, we describe a algorithm a that computes a function f mapping every instance of L prime

problem to some approximately constructed instance fx of the unknown problem which we are going to prove to be NP complete.

Now, we have to prove that this x the problem instance in L prime if and only if the corresponding mapping of that particular problem into an L into the problem L means the problem which we are going to prove to be NP complete; that means, we take a problem in L prime and we map it to another problem that is L and this particular mapping is to be done by an algorithm which runs in a polynomial time. So, so we have to choose we have to know that what problems are there. So, if you want to prove a particular problem a new problem to be NP complete we have to choose one of these particular problem, let us say circuit satisfiability problem and we reduce this circuit satisfiability problem to that new problem this direction is important otherwise the transitively it will not be able to reduce and hence it will not be the correct solution. So, from a known NP complete problem to a new problem we have to reduce in this particular direction and apply these steps to prove it to be NP hard.

(Refer Slide Time: 63:07)

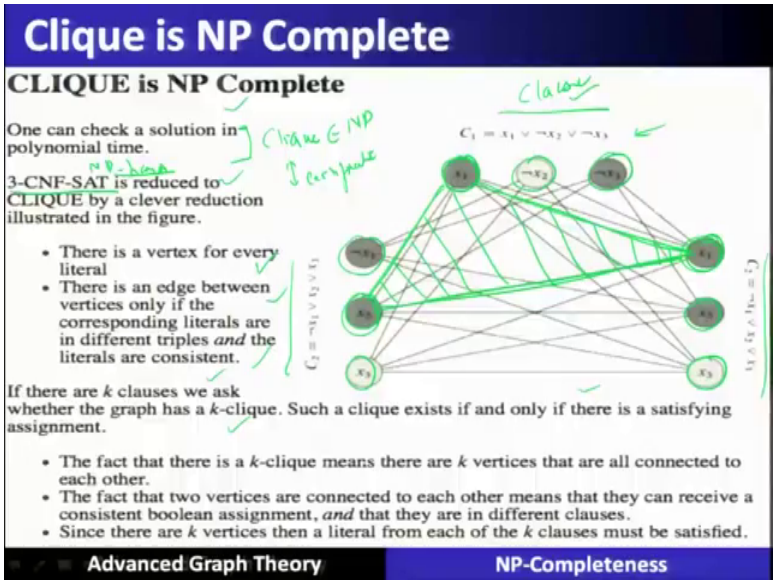


Once the problem is an NP and we have proved it to be NP hard and the problem becomes; let us take the example of the clique problem. So, a clique in an undirected graph is a subset of vertices such that each pair of these subset of set of vertices or connected by an edge that is they are mutually connected set that is called a clique. So, take for example, this is the graph and k 2 is basically a clique present all vertices if

include that also completes a clique. So, the clique of a maximum size we have searching. So, a clique problem is to find the clique of a maximum size this becomes an optimization problem. So, this can be converted into a decision version of a problem by asking whether a clique of a size k exists in the graph or not.

So, we are we are given a pair that is the graph and a value of k. So, we are asking a question whether the graph contains a clique of size k or not. So, this is a decision version of the problem which will answer say yes or no and that is called a clique problem.

(Refer Slide Time: 64:24)



Now, to prove that the clique is NP complete; so, first thing is we have to show that the clique is an NP; how we can show this we can give a certificate and check whether the certificate is basically a clique of that graph or not and that verification, we can do in a polynomial time. Hence, this can be done we can do this in a polynomial time, hence the clique is in NP. Now second problem is second part is that we have to prove that clique is NP hard to prove it to be NP hard. We have to select one known problem which is there in NP complete.

Let us choose a 3 CNF-SAT which is well known NP complete problem SAT is circuit satisfiability. So, we are given a bullion function that will basically represent a circuit and we will reduce this particular problem to the clique problem by a clever reduction which is shown here in this particular diagram now the bullion function is the

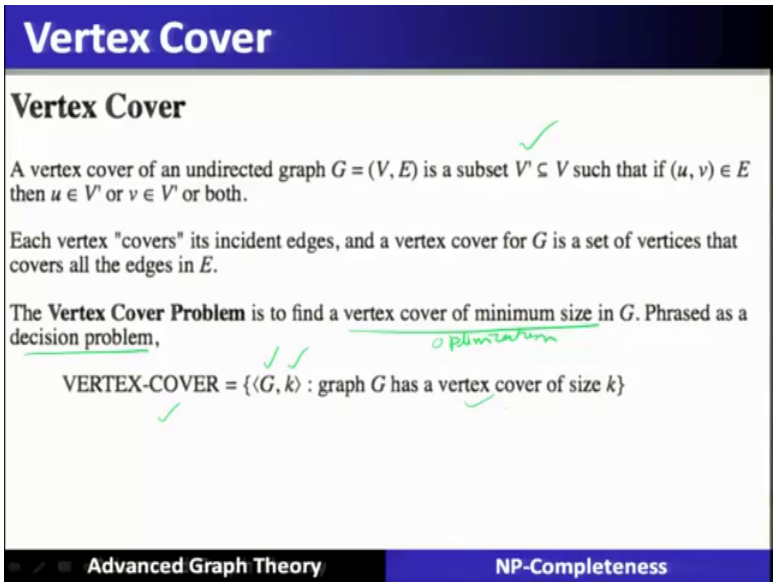
conjunction of disjunctions. So, the dis-junction are nothing, but they are the clause. So, let us say that it is that 3 CNF-SAT means every clause have has 3 literals x 1 and not of x 2 and not of x 3.

Similarly here also this clause C 2 is also having 3 literals and C 3 is also having 3 literals. So, each literal is represented as the node of a graph and now we will place the edges we will not place the edges between x 1 and x 1 not of x 1, we will not place the edges between the x 1 and not of x 1, we will place the edge between other parts like x 1 and x 2 x 1 and x 3 where there is no conflict, we can place an edge over this graph we obtain my in this graph, we will see that the solution to this particular bullion formula is there, if we basically select one variable from each particular clause and assign the truth value one.

So, this particular formula becomes true in that case and this is nothing, but a clique. So, if the formula is basically valid, then there exist a clique; that means, we have converted a clique or we have reduced into a clique of a graph. So, again let us recall that there is a vertex of every literal there is an edges between the vertices only if the corresponding literals are different in different triples and literals are consistent; that means, x 1 and x on bar or not consistent. So, there a k clauses we ask whether the graph has a k clique such a clique exist if and only there is a satisfying assignment the fact that there there is a k clique means there are k vertices that are all connected to each other the fact that two vertices are connected to each other means that they can receive the same consistent bullion assignment and they are in different clauses.

Since they are k vertices, then literal form each of k clauses must be satisfied. So, we have shown that clique is NP complete in this particular reduction method.

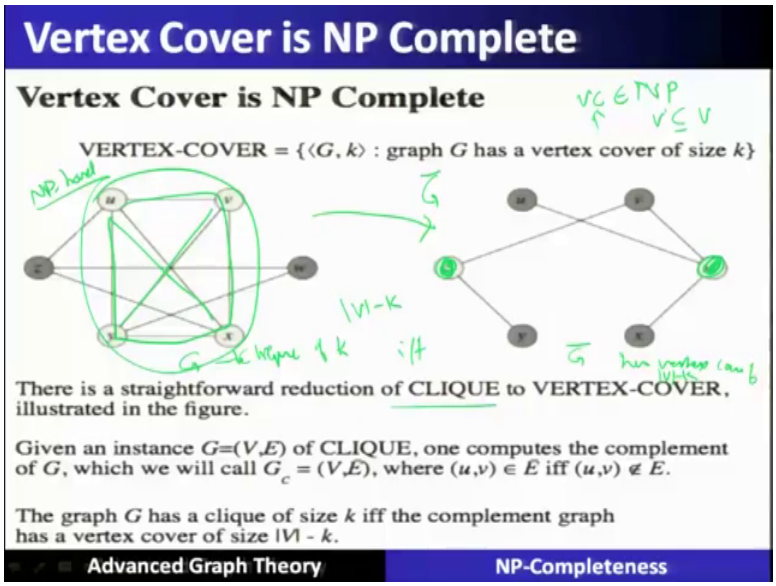
(Refer Slide Time: 68:30)



Vertex cover a vertex cover of an underrated graph is a subset of vertices such that if u and v is an edge of a graph then u is either u is basically taken up in the vertex cover B prime or v is taken up in the vertex cover or they are both taken in the vertex cover. So, the vertices covers its incident edges and the vertex cover of a graph is is the set of vertices that covers all the edges in that particular graph vertex cover problem is to find the vertex cover of a minimum size is this is a optimization version of a problem.

Now, the decision version of a problem would be that called vertex cover problem is nothing, but a graph is given and the k is given and the question is asking whether the graph as the vertex cover of size k.

(Refer Slide Time: 69:34)



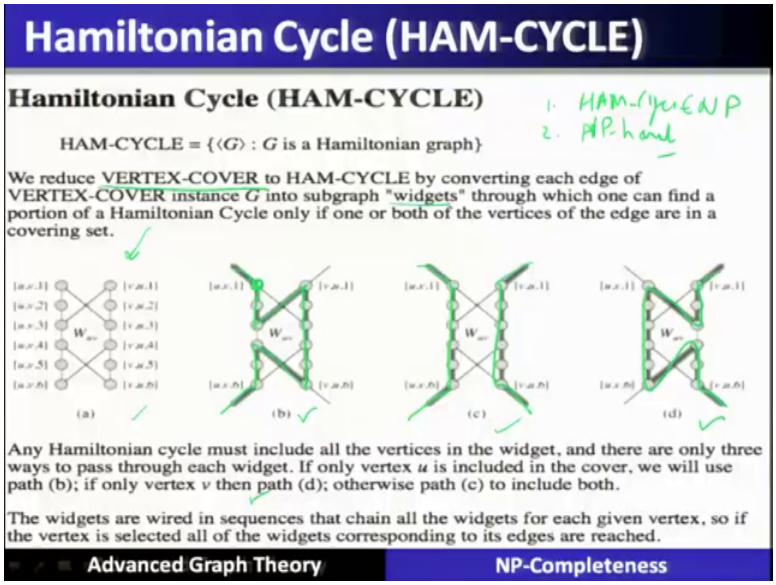
Now, the vertex cover is NP complete, let us see how we can prove this particular problem to be NP complete for that the first step that vertex cover is NP is quite easy. So, we give a certificate in the form of a v prime which is a subset of v and then we have to check in a polynomial time whether this v prime is the vertex cover of a graph or not that is it covers that is all the edges are incident in B prime or not that we can check in a polynomial time.

Now, the other parts remain that we have to prove that vertex cover is NP hard to prove that vertex cover is NP hard, we have to give a reduction that is the polynomial function which will convert a given NP complete problem to the vertex cover problem. So, let us choose a particular problem that is called a clique we have earlier proved that clique is NP complete. So, one NP complete problem we will take now we will reduce to a vertex cover problem now given an instance G V E of a clique one computes the compliment of a graph G if this G is given we compute a compliment of that particular graph which we call G compliment.

So, the graph G has a clique has a clique of size k if and only if the compliment has the vertex cover of size V minus k. So, here we can see that in G, there is a clique of size 4 all the vertices are connected we convert into a compliment of that particular graph and all these edges are missing. So, in the compliment graph, we will get the vertex cover. So, vertex cover will be of this particular size and this is nothing, but V minus if we take

V minus k this two the remaining will be that particular vertex cover in the original graph.

(Refer Slide Time: 72:18)



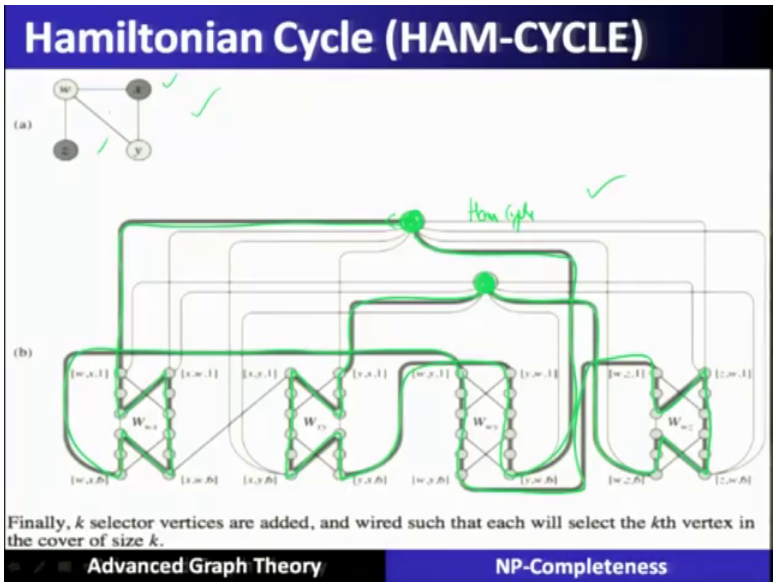
Now, we will see another problem called Hamiltonian cycle to prove that Hamiltonian cycle is NP complete we have to show that Hamiltonian cycle to be in NP; that means, if you give a certificate in the form of the cycle or a or a path you can check whether it is Hamiltonian cycle or not that can be done in the polynomial time second part, we have to prove that that it is in n it is NP hard for that we have to choose one problem which is already there in NP complete problem and we can reduce that particular problem to the Hamiltonian cycle problem by way of a polynomial time reduction or a mapping.

Now, let us choose the vertex cover a wel[l] a well know problem as NP complete problem and we will reduce to the Hamiltonian cycle problem by converting each edge of a vertex cover instance into a subgraph which is called a widgets through which one can find a portion of the Hamiltonian cycle only if one or both of the vertices of the edge are in the covering set. So, this is non-trivial problem to convert an instance of a graph into the sub graph of widgets which is shown over here in this particular figure. Now NP Hamiltonian cycle must include all the vertices in the widgets and there are only 3 ways to pass through each digit if only vertex u is included in the cover, we will use the path b; that means, it will enter from u and it will trace back on the other side of other partite side of the widgets then come back return to the to the same partite sets and exists from

if only vertex v then we have to; that means, if there is entry at vertex v and we have basically this particular widget d.

otherwise the path C to include this particular path; that means, we enter from u complete the widget of one side enter from v complete the widget from the other side the widgets are wired in sequences that chain all the widgets of each given vertex. So, if the vertex is selected all of the widgets corresponds to its edges are reached.

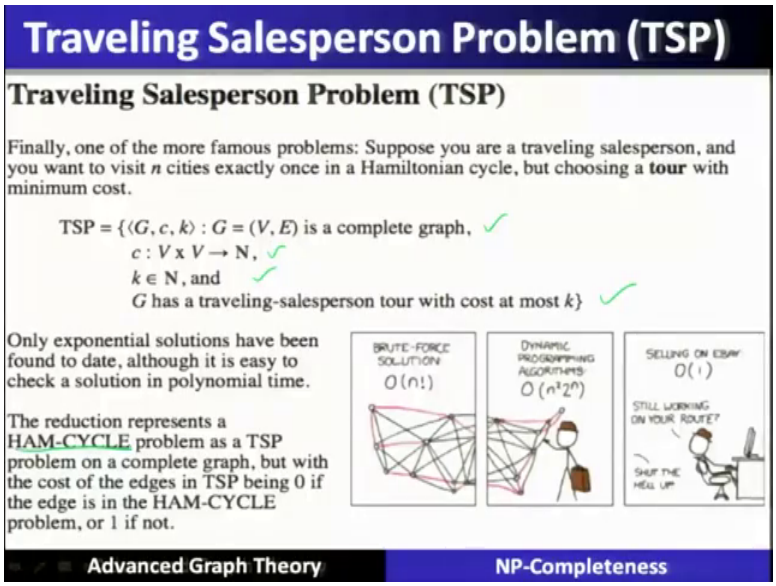
(Refer Slide Time: 75:01)



Let us take this example graph and this example graph is converted into into this particular graph of widgets. So, here we can see that if we select a vertex x 1 than we can enter through a particular widget W trace back all the widgets and then exits out from this particular widgets and enters into another set of widgets and enter into another set of widgets enter into another set of widgets go to another vertex of the vertex cover comes back and trace back all the other widgets and return back to the same points from where it has started this is the Hamiltonian cycle.

And these two vertices if there is a Hamiltonian cycle; that means, they are visited through these two vertices. So, there is a vertex cover. So, the vertex cover is converted into or reduced into a Hamiltonian cycle problem here with these particular reduction.

(Refer Slide Time: 76:19)



Another problem is that travelling salesman problem travelling salesman problem, we have to prove to be NP hard. So, finally, one of the famous problems suppose, you have travelling through across through different cities exactly once in a Hamiltonian cycle, but this particular tour should be of minimum cost.

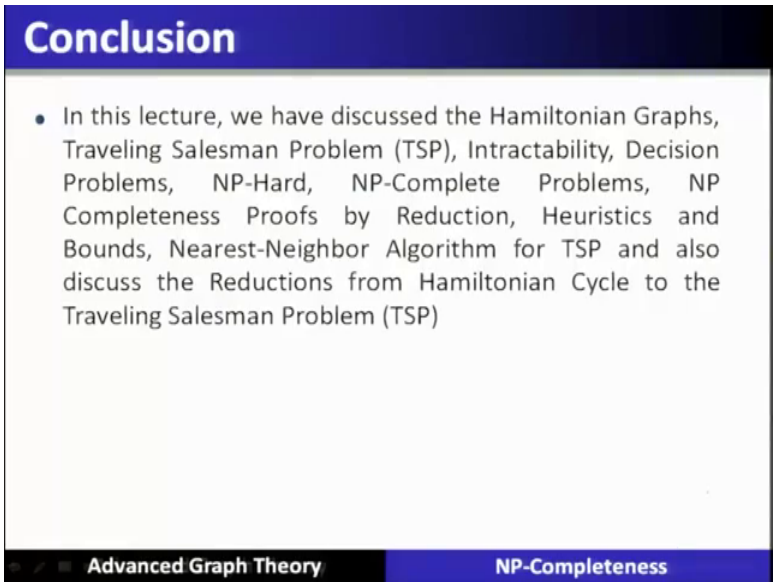
So, travelling salesman as I told you earlier requires a graph to be of complete graph and every edge will be having a weight to it and we want to find out a value k. So, that the tour should be having the cost with that as at most k value now here only exponential solution have been found today; that means, there is not good algorithm although it is easy to check the solution in a polynomial time the reduction represent a Hamiltonian cycle problem as the travelling salesman problem on a complete graph, but with the cost of edges in a travelling salesman problem being 0, if the edge is in Hamiltonian cycle and the problem or if it is one and if it is not there.

So, we can see that we have reduced this particular travelling salesman Hamiltonian cycle problem to a travelling salesman problem and we have proved that it is basically NP complete problem. Now as per as travelling salesman problem are concerned if you know that there are brute force solutions are exists; that means, we span we we find out all the different cycles and find out the cycle with a minimum cost and that will we exponential number of possibilities to explore and then find out the good solution; so, optimal solution. So, this is having a exponential time another way of solving travelling

salesman is through the dynamic programming, but the that that also requires the order of n square 2 is power n which is also an exponential size such space or the time is exponential.

So, the; so, if there is no good solution. So, there is a there is a cartoon which says that let us not go on tour and let us sit down; that means, it is going to take a lot of time to compute the minimum cost.

(Refer Slide Time: 79:31)



So, we have proved this NP complete problem as the travelling salesman problem. So, in this lecture, we have discussed Hamiltonian graphs travelling salesman intractability decision problems NP hard NP complete NP-Completeness proved by reduction heuristics and bounds nearest neighbor algorithms in travelling salesman and also discussed reductions from Hamiltonian cycle to the travelling salesman problem.

Thank you.