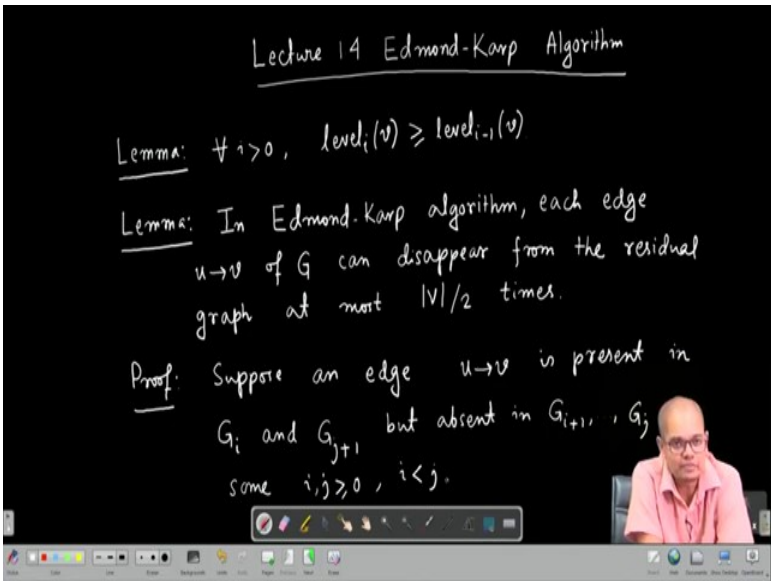
Selected Topics in Algorithm
Prof. Palash Dey
Department of Computer Science and Engineering
Indian Institute of Technology - Kharagpur

Lecture - 04
Edmond-Karp Algorithm (Contd.)

Welcome, in the last class we have started studying Edmonds-Karp Algorithm and we were doing correctness analysis so, we will finish that today.

(Refer Slide Time: 00:36)

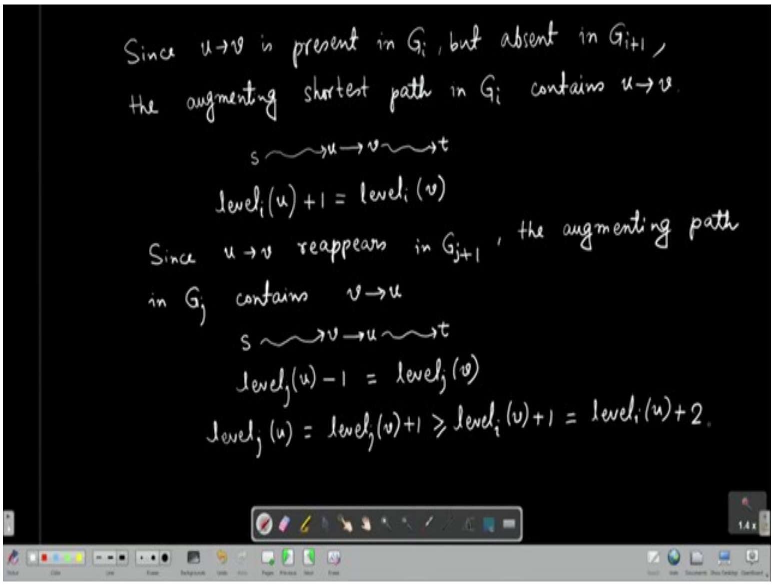


So, in the last class we have proved an important lemma that for all integer i greater than 0 the level i of v, this is the unweighted shortest path for distance of $v \in G_i$ from is this is greater than equal to level i − 1 of v. So, it says that in every iteration in the residual graph, the distance the unweighted shortest path distance of v from is keeps on increasing, keeps on it never decreases.

So, using this we will bound how many times a particular edge can be can disappear? And using that we will bound the total number of iterations of the algorithm. So, let me state the next lemma in Edmond-Karp algorithm each edge u to v of G can disappear from the residual graph at most size of $|V|/2$ times. Proof, so suppose and edge u to v is present in G_i and G_{j+1} but absent in G_{i+1} dot dot dot, G_j for some i, j greater than equal to 0, i is less than j.

So, suppose this edge u to v was present in G_i but from G_{i+1} onwards till G_j it was absent and it again reappeared in G_{j+1} in the residual graph. Let us recall when does edge can disappear? If it is saturated if it is current flow value is it is capacity then that edge is becomes absent in the residual graph.

(Refer Slide Time: 05:43)

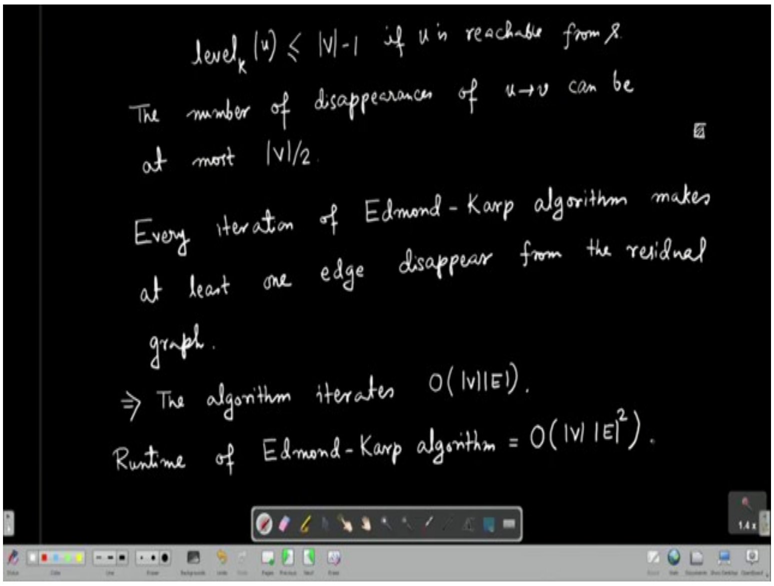


So, let us see so, since u to v is present in G_i but absent in G_{i+1} this can only happen if the augmenting shortest path picked in G_i contains edge u to v. The augmenting shortest path in G_i contains u to v this edge. So, the shortest path looks like here is s and it uses u to v edge and then it reaches t. So, we can conclude that the level i of u + 1, the unweighted shortest path of distance of u is 1 less than the unweighted shortest path of distance v which is equal to level i of v.

But this edge u to v reappears in G_j , so, since u to v reappears in G_j in particular in G_{j+1} in particular it is it was absent in G_j and it was it reappeared in G_{j+1} . The only way it is possible that the augmenting path in G_j contains v to u that is the shortest path in G_j from s, it first reaches v then it uses the edge u and then it reaches t. So, the level j of u − 1 is level j of v. So, now let us see level j of u is level j of v + 1 which is at least as i as level i of v + 1.

Because level only increases and this is equal to level i of v is nothing but level i of u + 1. So, in other words between every disappearance and reappearance of the edge u to v the level of the vertex u increases by at least to the distance. The shortest distance the unweighted shortest path distance of u increases by at least 2.

(Refer Slide Time: 11:01)

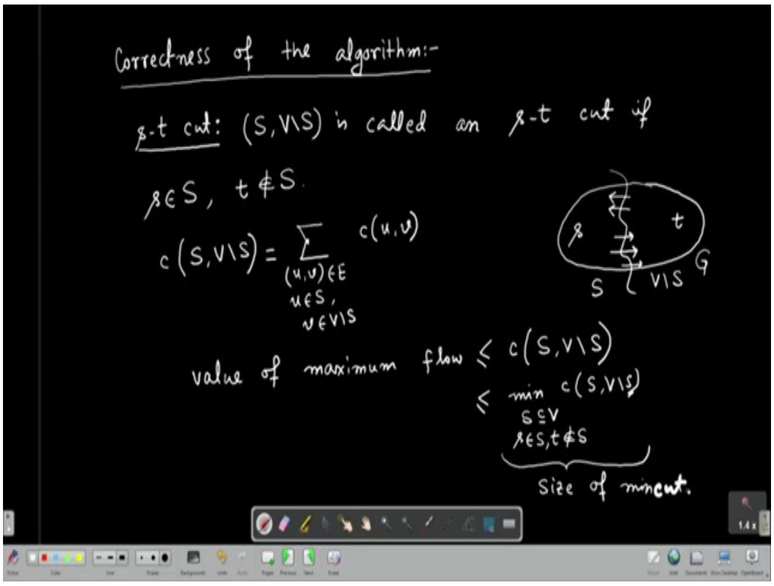


But the unweighted with the level I do not know level k of u is less than equal to size of V − 1. The graph is connected and the level of any vertex the unweighted shortest path distance is at most V − 1. So, the number of disappearance says of this edge u to v if u is reachable from s. Because you know in the residual graph it may be possible that it is not reachable but whenever the edge u v disappears at that time u is reachable.

Because the shortest path passes through you the shortest edge to t path so, it is reachable from s. So, the number of disappearance of u can be at most size of $|V|/2$. Now, an observation is that every iteration of Edmond-Karp algorithm makes at least one edge disappear from the residual graph. Hence, so this implies that the algorithm iterates at most $|V|/2$ times E many iterations so, big O of size of V times size of E iterations.

Now, each iteration can be executed by in V + E time by finding the unweighted shortest path from s-t which can be computed using a BFS. So, the runtime of Edmond-Karp algorithm is big O of V times E iterations and each iterations take order V + E time which is same as order E. The runtime is order V times order E square. So this shows that the runtime is polynomial but how about correctness?

(Refer Slide Time: 15:42)



The algorithm holds the Ford-Fulkerson algorithm holds if there is a there is no s t path in the residual graph. Now, why does at the termination, the flow we have is the maximum flow. So, correctness of the algorithm so algorithm holds if when there is no s-t path in the residual graph. Now, you see that we will show some lower bound so, let us define the notion of s-t cut, S, V − S is called an s-t cut.

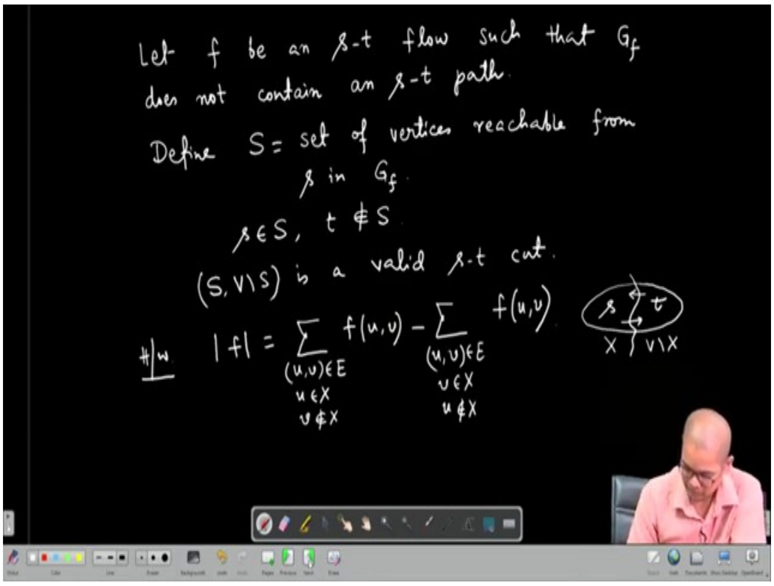
If the vertex S belongs to the set S and the vertex t sync does not belong to the set S. So, what is the let us define the capacity of the cut? So, c of S, V − S this is see, if you look at this cut pictorially here is the graph G and this is the set S this is the set V − S here is the source vertex, here is the sink vertex. So, each cut imposes a upper bound on the maximum flow value.

So, there are two kinds of edges one edge goes from S to V − S and the other edges goes from V − S to S. So, the capacity of the cut is the sum of the capacities of the edges which go from S to V − S. So, this is sum of u, v edge u, v such that u belongs to S, v belongs to V − S and this is the capacity c u, v. Now, you see that any flow from s-t has to cross this cut and this capacity of this cut becomes a upper bound.

So, value of maximum flow is less than equal to c of s of V − S. So, each s-t cut imposes a upper bound on the value of the maximum flow and hence this value of the maximum flow is less than equal to minimum over s capacity of S, V − S, S is a subset of V and small s is contained in S, t is not content in S. What we will show is that at the termination of the Ford-Fulkerson algorithm and thus of the Edmond-Karp algorithm.

The value of the maximum flow is actually coincides with this minimum this particular quantity is called size of min cut. So, obviously if any flow value matches with the size of the min cut then that has to be a maximum flow.

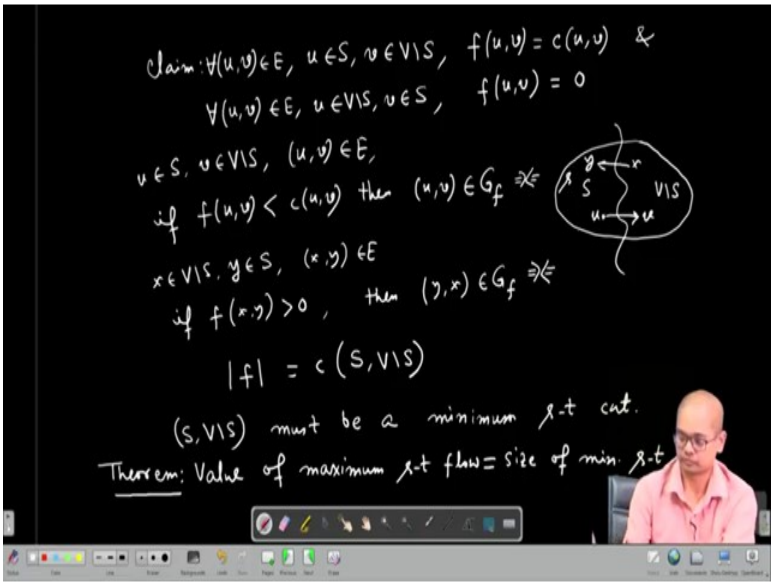
(Refer Slide Time: 21:19)



So, let or we will show that let f be a flow an s-t flow such that G_f does not contain an s-t path. Define capital S is the set of vertices reachable from s in G_f, clearly the source belongs to s and sink does not belong to S because there is no s-t path that is by assumption. So, S, V − S is a valid cut now, here is one easy homework is that the value of any flow is if you take any cut, any s-t cut here is X here is V − X.

Here in X there is source and t is in V − X then this is the sum of the flow values $(u,v) \in E$, $u \in X$, $v \notin X$. These are X to V − X edges, the flow values of these edges minus the flow value of opposite edges, sum of the flow values of opposite edges. So, now S, V − S is a valid s-t cut.

(Refer Slide Time: 24:55)



So, claim you for every edge for all edge $(u,v)\in E$, u belongs to S, v belongs to V − S then $f(u,v)$ must be equal to $c(u,v)$. And so, the edges from S to V − S they are completely saturated and edges from opposite direction $(u,v)\in E$, u belongs to V − S, v belongs to S, f of u, v is 0. So, the V − S to is those sort of edges does not carry any flow and the other edges, these are forward edges they are completely full.

Indeed, this has to be the case because let us do the proof by picture, very easy. So, let us take an edge from left to right S to V − S and if it is not saturated then this suppose this u and v and if this edge is not saturated then this edge. If so, u belongs to S, v belongs to V − S u, v is an edge if f of u, v is less than c u, v. That means if this edge is not saturated then this edge u, v is present in G_f .

But now, let us recall how do we define S? Capital S is the set of vertices reachable from source vertex s in G_f and if this edge is present and if u is reachable then v is also reachable then how come v does not belong to s so that is a contradiction. That means this sort of edges u to v must be saturated, same with the reverse phases. So, suppose there is the edge from x to y.

So, x belongs to V − S, y belongs to S, x, y is an edge if f of x, y is greater than 0 if it is not 0 then this edge, the reverse edge y to x must be present in G_f . But y is reachable from s and if y, x y to x edge is there how come x is not reachable so, again we have a contradiction. So, what we have shown is that the value of f is equal to the size of the capacity of this cut and we know that the size of the maximum flow could be is upper bounded by every cut in particular the size of the min cut.

So, this S, V − S then must be a minimum s-t cut which proves the celebrated max flow mean cut theorem. Value of maximum s-t flow is same as size of minimum s-t cut. Not only that using this Ford-Fulkerson method, we can or in Edmond-Karp algorithm, we can compute minimum s-t cut. So, let us stop here.