Second Level Algorithms

Prof. Palash Dey

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Week – 07

Lecture 31

Welcome to the 31st lecture of the second-level algorithms course. In the last lecture, we started proving the max-flow min-cut theorem. So, let us continue the proof of that theorem. So, proof of max-flow min-cut theorem.

We have already proved that the max s-t flow value is less than or equal to the minimum capacity of any s-t cut. So, in this lecture, we will show the other direction. So, let

any max s-t flow; then there is no s-to-t path in the residual graph G_f. Recall this was the optimality criterion of a flow being a maximum s-t flow that we have used in many of our algorithms. That means, if I look at the residual graph G_f, there is no s-to-t path because f is a maximum s-t flow. So, here we consider the following subset A of vertices.

A is the set of vertices such that v is reachable from S in G_f. of course, S belongs to A because S is reachable from S and because S is a maximum S-T flow from optimality criteria we know that T is not reachable from S. So, T belongs to V minus A. So, this is the set A ok. So, we claim that for every edge u v in the input graph G, not in the residual graph, such that u belongs to A and v belongs to $V \setminus A$. The flow value in that edge must be the same as the capacity of the edge.

That means all such edges are saturated ok. So, there is the input graph G and if I look at this S-T cut $(A, V \setminus A)$ defined using G_f and look at any edge which go from A side to V minus A side such edges must be saturated. that should be the case indeed otherwise if there is an edge u to v where u belongs to A and v belongs to $V \setminus A$ which is not saturated.

then the edge uv must be present in the residual graph. However, this implies that v also should belong to A, contradicting our assumption that v belongs to $V \setminus A$. Similarly, if I

consider any edge say x to y going from b minus a side to a side. must carry no flow that is f should be 0.

Indeed, this should be the case otherwise the edge y to x will be present in the residual graph. that means, in the residual graph G_f y to x edge is present. However, this implies that x is also reachable from S and thus x should belong to A.

this contradicts our assumption that x belongs to $V \setminus A$. ok now in the forward direction one direction of the proof of max flow min cut theorem in the last class we have seen that value of f is total flow going out of A minus total flow going into A. Now, all the edges going from outside of A to inside of A, they carry 0. So, this quantity is 0 and all the edges from A side to $V \setminus A$ side, they are fully saturated.

So, the first term is the capacity of the s t cut a comma b minus a. So, it is one particular s-t cut. So, we conclude that value of f is greater than equal to minimum over all s-t cuts because it is equal to one particular s-t cut capacity of one particular s-t cut and minimum over all such s-t cuts the capacity can be even less. So, this is minimum over s-t cut.

ok. Before we have shown that value of f is less than equal to minimum capacity of a sticker, now we show greater than equal to. So, we conclude that value of f is equal to minimum capacity of any s t cut, but f was a maximum s-t flow. So, we conclude that value of a maximum S-T flow is the capacity of minimum S-T cut. Moreover, such a minimum S-T cut can be computed in polynomial time. So, this concludes the proof of the max flow min cut theorem. Moreover, a minimum s t cut can be computed in polynomial time as follows.

We use any polynomial time algorithms to compute a maximum s-t flow if then a minimum s t cut is as we have seen in the proof that here value of f is the capacity the cut $(A, V \setminus A)$. So, then a minimum S-T cut is $(A, V \setminus A)$, where A is the set of vertices which are reachable from S in the residual graph. Next we will see another important application of max flow problem in finding the maximum bipartite matching.

So, what is this problem? For that, let us define or recall the notion of a bipartite graph. It is called a bipartite graph. A graph G is called a bipartite graph if the vertex set V can be partitioned into two sets say L and R, such that every edge of G has one endpoint in L and another endpoint in R. Equivalently, both L and R are independent sets. What is an independent set? An independent set is a set I of vertices such that every edge of G has at most one endpoint in I, okay. So, these are bipartite graphs. If I draw pictorially, this is

the bipartition of the vertices. Let us call them L and R, and all the edges are cross edges. This is how a bipartite graph looks like.

Now, let us recall what a matching is. A matching is a set of edges in a graph. with no shared endpoints. For example, let us consider a graph, say A B C D. So, let us consider this graph, and any set of edges with no shared endpoints is a matching.

So, these are some of the matchings. For example, if I take any singleton edge, that is a matching. If I take, say, AB and EF, this is also a matching. No two edges share an endpoint. For example, AB, CD, EF is also a matching. Examples of not matching: for example, if I take the edge AB and A this is not a matching because the edges AB and AF share an endpoint. For example, if I take AB, CD, and DE, this is not a matching because in this set, there exist two edges which share an endpoint, namely CD and DE. So, what is the bipartite matching problem?

In the bipartite matching problem, In the maximum bipartite matching problem. We are given a bipartite graph, and we need to compute a maximum cardinality matching of that bipartite graph.

So, in the next lecture, we will see how using a maximum s-t flow problem, we can solve —we can get a polynomial-time algorithm for maximum cardinality bipartite matching, okay? So, let us stop here. Thank you.