

## Discussion 8

---

1. You have successfully computed a maximum s-t flow  $f$  for a network  $G = (V; E)$  with integer edge capacities. Your boss now gives you another network  $G'$  that is identical to  $G$  except that the capacity of exactly one edge is decreased by one. You are also explicitly given the edge whose capacity was changed. Describe how you can compute a maximum flow for  $G'$  in  $O(|V| + |E|)$  time.

Solution:

Say the edge  $vu$  the edge that has lost capacity, then

- Find a path in  $G$  from  $s$  to  $v$  on edges that are carrying some flow
- Find a path in  $G$  from  $u$  to  $t$  on edges that are carrying some flow
- Reduce flow by unit on edges that are on the path  $s \rightarrow v - u \rightarrow t$ . Since all these edges are carrying some flow, then the results will be a new valid flow  $f'$
- Construct  $G_{f'}$
- Try to find a path in  $G_{f'}$  from  $s$  to  $t$ . If there is such a path then push (augment) one more unit of flow on that path. The resulting flow will be our max flow. If there is no such path then  $f'$  is our max flow.

Complexity: all the above steps can be done in linear time, therefore the whole process takes linear time.

2. You need to transport iron-ore from the mine to the factory. We would like to determine how long it takes to transport. For this problem, you are given a graph representing the road network of cities, with a list of  $k$  of its vertices  $(t_1, t_2, \dots, t_k)$  which are designated as factories, and one vertex  $S$  (the iron-ore mine) where all the ore is present.

We are also given the following:

- Road Capacities (amount of iron that can be transported per minute) for each road (edges) between the cities (vertices).
- Factory Capacities (amount of iron that can be received per minute) for each factory (at  $t_1, t_2, \dots, t_k$ )
- The amount of ore to be transported from the mine,  $C$

Give a polynomial-time algorithm to determine the minimum amount of time necessary to transport and receive all the iron-ore at factories.

Solution:

Construct a flow network  $G$  as follows:

- Choose  $S$  as the source
- Create a new sink node called  $t$ . Add directed edges from each  $t_i$  to  $t$  with edge capacity associated with factory  $i$ 's receiving capacity.
- Replace each undirected edge (roads) with two directed edges with the same capacity as the original undirected edge

Find max flow  $f$  in  $G$  using a polynomial time max flow algorithm. The time required to move all  $C$  units of iron-ore to the factories will be  $T = C/v(f)$

3. In a daring burglary, someone attempted to steal all the candy bars from the CS department. Luckily, he was quickly detected, and now, the course staff and students will have to keep him from escaping from campus. In order to do so, they can be deployed to monitor strategic routes.

More formally, we can think of the USC campus as a graph, in which the nodes are locations, and edges are pathways or corridors. One of the nodes (the instructor's office) is the burglar's starting point, and several nodes (the USC gates) are the escape points — if the burglar reaches any one of those, the candy bars will be gone forever. Students and staff can be placed to monitor the edges. As it is hard to hide that many candy bars, the burglar cannot pass by a monitored edge undetected.

Give an algorithm to compute the minimum number of students/staff needed to ensure that the burglar cannot reach any escape points undetected (you don't need to output the corresponding assignment for students — the number is enough). As input, the algorithm takes the graph  $G = (V, E)$  representing the USC campus, the starting point  $s$ , and a set of escape points  $P \subseteq V$ . Prove that your algorithm is correct and runs in polynomial time.

### Solution:

This is a min-cut problem. I.e. we need to find the minimum number of edges in the network whose removal will disconnect source (CS dept) from the sink (all exit points). Here are the steps to create the flow network  $G$ :

- CS Dept will be the source  $s$
- Create a new sink node  $t$  and connected all exit points to  $t$  with a directed edge having a very large capacity (at least  $m$ ). The reason for the large capacity is that we don't want any of these new edges to appear on the min cut since these are not really representing pathways or corridors that could be monitored.
- Replace each undirected edges (pathways) with two directed edges each with capacity 1

We then find max flow  $f$  in  $G$ .  $v(f)$  will be the minimum number of students/staff needed to ensure that the burglar cannot reach any escape points undetected.

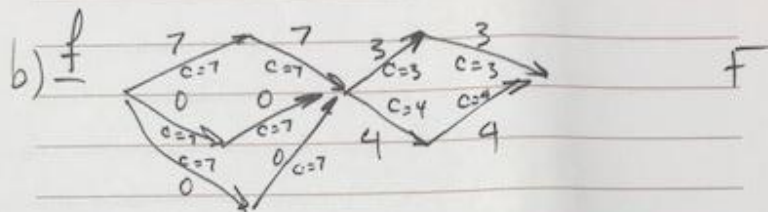
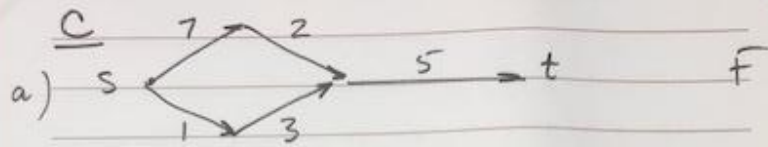
Complexity: even if we use Ford-Fulkerson to solve the max flow problem in  $G$ , our solution will be polynomial time since edge capacities for edges leaving  $S$  are all 1 and there are only  $O(n)$  of them. So the complexity of the FF algorithm will be  $O(nm)$  for this particular problem.

If we are also interested in finding the exact pathways to assign the staff to, (having max flow  $f$ ) we will follow the process described in lecture to find a min cut in  $G$ .

4. We define a most vital edge of a network as an edge whose deletion causes the largest decrease in the maximum  $s$ - $t$ -flow value. Let  $f$  be an arbitrary maximum  $s$ - $t$ -flow. Either prove the following claims or show through counterexamples that they are false:

- (a) A most vital edge is an edge  $e$  with the maximum value of  $c(e)$ .
- (b) A most vital edge is an edge  $e$  with the maximum value of  $f(e)$ .
- (c) A most vital edge is an edge  $e$  with the maximum value of  $f(e)$  among edges belonging to some minimum cut.
- (d) An edge that does not belong to any minimum cut cannot be a most vital edge.
- (e) A network can contain only one most vital edge.

All are false. See counterexamples below.



c) Same as a  $F$

d) Same as a  $F$

e)  $\rightarrow$   $F$