

(2) (10 points) Design a linear-time algorithm that recomputes a maximum flow in a network after the capacity of one edge is decremented. The input to the algorithm consists of:

- a flow network  $G$  with integer edge capacities,
- a integer flow  $f$  that is a maximum flow in  $G$ ,
- an edge  $e = (u, v)$ .

If  $G'$  denotes the flow network obtained from  $G$  by decreasing the capacity of edge  $e$  to  $c(u, v) - 1$ , then your algorithm should output a maximum flow in  $G'$ , and it should run in time  $O(m + n)$ , where  $m$  and  $n$  are the number of edges and vertices in  $G$ , respectively.

## Solution

Let the edge whose capacity is decremented by be  $e = u, v$ . If the edge wasn't fully saturated which means that  $f(e) \neq c(e)$ , then the maximum flow remains unaltered by this operation.

If  $f(e) = c(e)$ , then we have the following case -

the outgoing flow on node  $u$  will be one less than the incoming flow at node  $u$ . Similarly the outgoing flow on node  $v$  will be one more than the incoming flow on  $v$ . Thus we will build the residual graph  $G'$  from  $G$  and look for a path from  $u$  to  $v$  so that the flow can be increased by one. If we find such a path, we will augment the maximum flow  $f$  with this path. This will make sure that the flow property at each node is satisfied.

If we don't find such a path, then we must reduce the flow from  $s$  to  $u$  and from  $v$  to  $t$  with one unit. We will do this by finding a path from  $u$  to  $s$  and from  $t$  to  $v$ . Then we will increment the flow on the edges on these paths by 1 ( we do this because these paths are in the opposite direction). Also we are guaranteed to find such paths since we had a flow from  $s$  to  $t$  via  $u-v$ . This operation will give us the new maximum flow  $f$ .

## Proof of correctness

The proof of correctness comes from the properties of residual graph. Firstly maximum flow can never increase by decrementing an edge simply because we are not increasing any values in the residual graph.

If the edge is not saturated in the max flow then decrementing it does not change the maximum flow simply because this operation does not add any paths from  $s$ - $t$ .

In the other case, it's evident that if we can find a path  $u-v$  in the residual graph then we can direct the missing 1 unit of flow along that path and our maximum flow value would not change and we can find the new maximum flow by augmenting the existing max flow  $f$  with this path. When such a path does not exist, the max flow value will decrease and we find paths  $u - s$  and  $t - v$  so that we can maintain the flow properties at all the nodes. The new flow vector can be found by augmenting these two paths to the existing flow vector  $f$ .

## Running Time

The slowest step in this algorithm is finding a path from  $u - v$  which can take  $O(m + n)$ . If we don't find such a path we look for paths from  $u$  to  $s$  and from  $t$  to  $v$  which takes  $O(m + n)$  again. It takes  $O(m + n)$  to build the residual graph as mentioned in the class. Hence the total running time is  $O(m + n)$