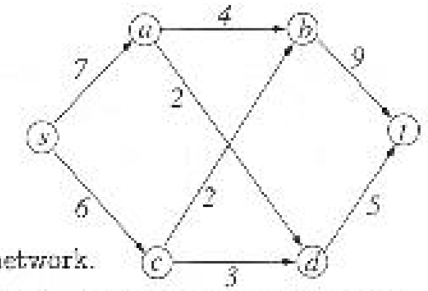
1. [30 points = 5 + 5 + 5 + 5 + 10] NETWORK FLOW

Consider the following flow network G where the edge capacities are specified and the start and the target nodes are labeled as s and t, respectively.



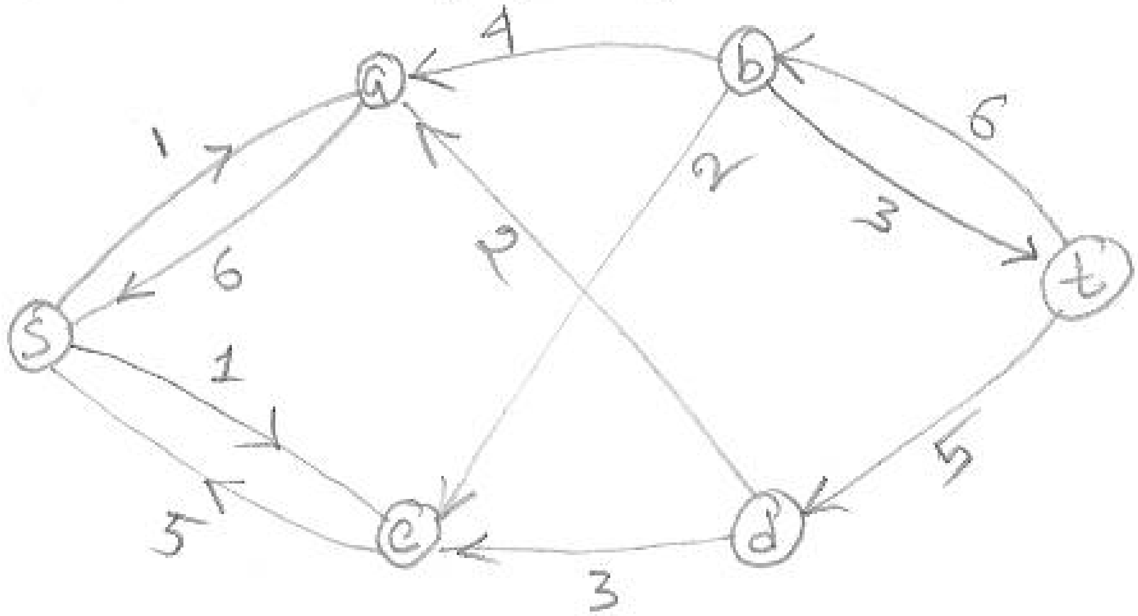
(a) Find the maximum flow f and a minimum cut on this network.

(b) Draw the residual graph G_f (along with its edge capacities), where f is the maximum flow found in part (a).

(a) Maximum Flow, f = 11

Minimum cut = {s, a, c} {b, d, t}

(b) Residual Graph, G_f



(c) An edge of a network is called a *bottleneck edge* if increasing its capacity results in an increase in the maximum flow. List all bottleneck edges in the above network G.

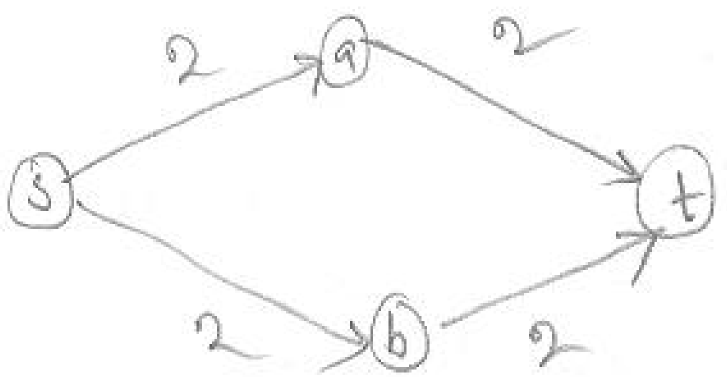
(d) Give a simple example (containing at most four nodes) of a network which has no bottleneck edges.

(c) Following Edges are bottleneck Edges.

→ (a,b) and (c,b)

[If you have reported some other edges, points are deducted, since it implies that you have not understood the concept of bottleneck edges].

(d)



(e) Design and analyze an efficient algorithm to identify all bottleneck edges in a network. (Hint: Start by running the usual network flow algorithm, and then examine the residual graph.)

step1:- Run the network flow algorithm and obtain the residual network.
Since, this residual graph is obtained after maximum flow, the residual network will not contain a path from source to sink.

step2: Run BFS, algorithm on residual graph from source vertex and make a set, S, of vertices reachable from, S.

step3: Run a second BFS algorithm after reversing all the edges and from sink. Collect all the vertéces, whice are reachable from sink. Let's call this set T.
Basically T contains all vertices from which sink can be reached.

Step 4: Now, for each directed edge (u,v) check, $u \in S$ and $v \in T$.

If this condition holds, then report this edge as bottle neck edges.

Time Complexity

→ time to run Network flow + 2 BFS + $O(E)$

$= O(VE^2) + O(V+E) + O(E)$