Homework 7

Thomas Kim tsk389 51835

David Munoz dam2989 51840

CS331 Algorithms and Complexity

Problem Q1(a). Prove or disprove

Claim: Consider an implementation of Ford-Fulkerson algorithm which does not create any backward edges in the residual graph. It is claimed that, there exists a constant $\alpha > 0$, such that for any flow network G, this modified implementation is guaranteed to find a flow of value at least α times the maximum-flow value in G **False**

Proof. A method of constructing a graph in which a particular execution of the above algorithm can produce a "max" flow of 1 whereas the max flow of the graph is an arbitrary positive integer.
First, begin with a graph consisting of 3 nodes, s, a, t, as shown in **Figure A**
The only possible flow of this graph generated by the above algorithm is trivially $f = 1 = \frac{1}{1}$
Next, add an additional edge b, with three additional edges: (s, b) with capacity 1, (b, a) with capacity 1, and (b, t) with capacity 1. This is shown in **Figure B**
The max flow of this graph is $f = 2$. Suppose the above algorithm begins by choosing path (s, b, a, t). The residual graph would contain only edges $(s, a), (b, t)$, so there exists no path from $s - t$
Therefore, the flow returned by the above algorithm is $f = 1 < 2$
Now, for $i = 2...\infty$ do: Add 3 additional nodes x, y, z. Let the current sink be denoted as w.
Node x will be the new sink t. An edge, (w, x) will be added with capacity i (the previous max flow).
Node y will have edges (w, y) with capacity 2, and (y, z) with capacity 1.
Node z will have edges (s, z) with capacity 1, and (z, y) with capacity 1.
This new graph has a max flow of $i + 1$. Simply take the previous max flow, send i flow over the edge (w, x), and 1 flow along the path (s, z, y, t)
However, suppose the algorithm found a "max" flow of value 1 in the previous iteration (which is trivially possible by induction on i).
Take the path used by this "max" flow and add the edges (w, y) with flow 1, and (y, x) with flow 1 to the path.
This new path, when explored first by the algorithm, causes the residual graph to have no path from s to t, as there exists no path from z to x since (y, x) is removed, and by induction the original graph similarly contains no path from any node $n \neq b$ to w, and adding x, y, z does not create any new paths from nodes in the original graph to w.
2 additional iterations of this construction are shown below in figures **C, D** **Figure A**

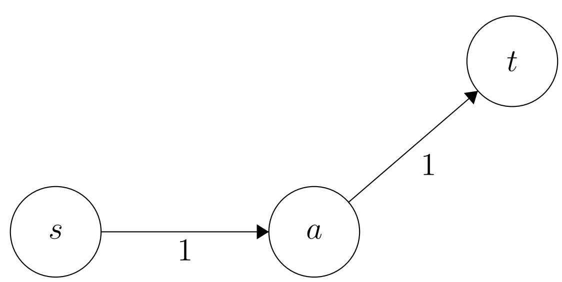


Figure B

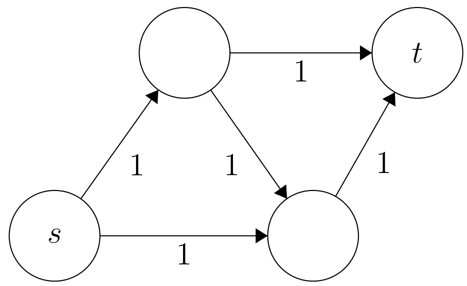


Figure C

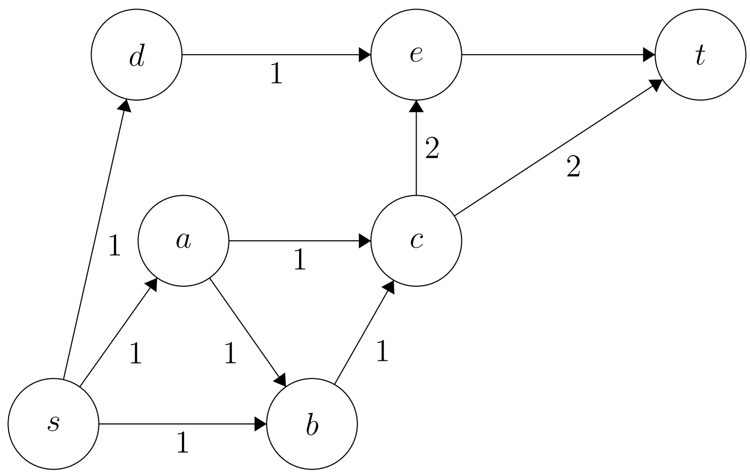
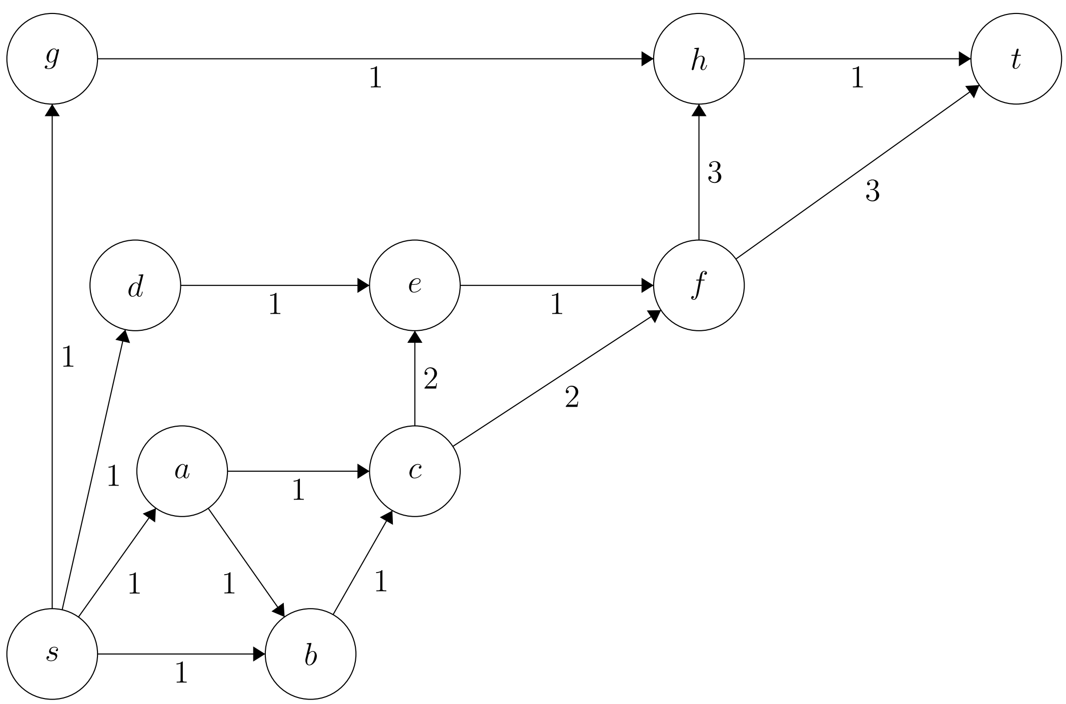


Figure D

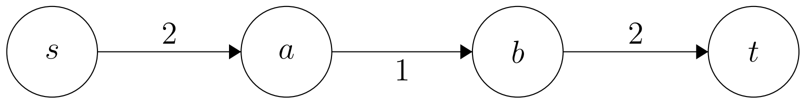


$\square$

Problem Q1(b). Consider a network $G = (V, E)$ with a source s, a sink t, and a capacity c_e on every edge $e \in E$.

Claim: if f is a maximum $s-t$ flow in G, then f either saturates every edge out of s or saturates every edge into t.
False

Proof.

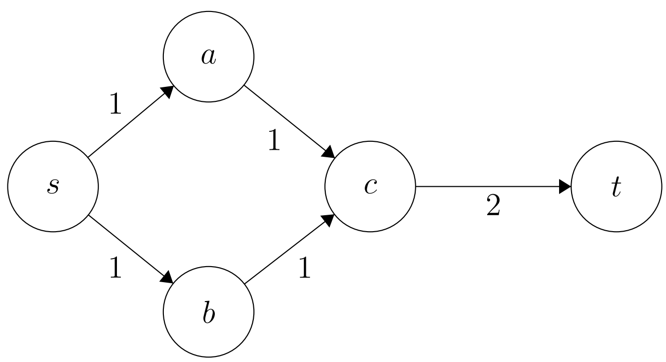


The min cut is $A = \{s, a\}, B = \{b, t\}$. The value of this cut is 1, which implies the max flow $f = 1$. This does not saturate the edge leaving s or the edge entering t.

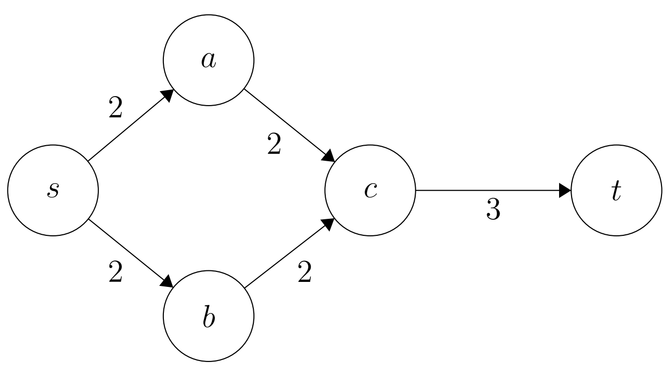
$\square$

Problem Q1(c). Consider a network $G = (V, E)$ with a source s, a sink t, and a capacity c_e on every edge $e \in E$. Let A, B be a minimum $s-t$ cut with respect to the capacities c_e.
Claim: If we add 1 to every capacity, namely $\forall e \in E, c'_e = c_e + 1 \implies A, B$ is still a minimum $s-t$ cut with respect to c'_e
False

Proof. Let G be the graph as pictured below:



Note that a min cut in G is $A = \{s, a, b\}, B = \{c, t\}$ with a value of 2
Now consider G'



Note that the cut $A = \{s, a, b\}, B = \{c, t\}$ now has a value of 4, whereas the cut $A' = \{s, a, b, c\}, B' = \{t\}$ has a value of 3

$\square$

Problem Q2(a).

Problem Q2(b).