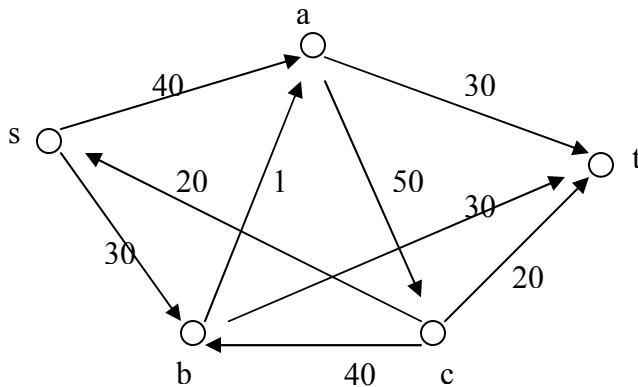


CSE 5311.004 Fall 2005
Exercise Set Weeks 9, 10

1. Run the Edmonds-Karp algorithm on the following network



2. In the above problem, if you ran the Ford-Fulkerson algorithm instead of the Edmonds-Karp algorithm, give an upper bound on the number of iterations your algorithm will make in the worst case.
3. Solve problem 26.2-9 from the textbook concerning how to determine **edge-connectivity** of an undirected graph using network flow.
4. Given two vertices s and t of an undirected graph, show how to use network flow to determine whether there are at least two paths between s and t that do not share any common intermediate edge (the paths are allowed to share common vertices).
5. Imagine you had not been taught Network Flows. Try to design an algorithm for computing a bipartite matching. Prove/disprove that it will compute the maximum bipartite matching.
6. Show the execution of the Longest Common Subsequence algorithm on the following two strings:
A B C A B D C D B A
B A C D D C A B
7. Given a single sequence of numbers, design an algorithm to find the longest monotonically increasing subsequence. For example, in the sequence

2 3 1 4 3 7 5 8

the longest monotonically increasing subsequence is

2 3 4 5 8

What is the running time of your algorithm?

8. Given two sequences of length m and n each, we studied in class how we can implement LCSS in $O(m \cdot n)$ time using $O(\min\{m, n\})$ space. Can you design an LCSS algorithm that only uses $O(1)$ space?

9. Given two sequences $S1$ and $S2$, design and analyze an algorithm to find the longest common **substring** between $S1$ and $S2$ (a substring is a subsequence which cannot skip intermediate characters, e.g. CABD is a substring of ABCABDCDBA).