University of Washington
Department of Computer Science and Engineering
CSEP 521, Winter 2013

March 15, 2013

Final Exam, Monday, March 18, 2013

NAME: ____________________________

Instructions:

- Closed book, closed notes, no calculators
- Time limit: Two hours
- Answer the problems on the exam paper.
- If you need extra space use the back of a page
- Problems are not of equal difficulty, if you get stuck on a problem, move on.

1	/15
2	/20
3	/15
4	/10
5	/20
6	/15
7	/15
8	/15
9	/15
10	/10
Total	/135

Problem 1 (15 points)Maximal Independent Set:

Let $G = (V, E)$ be an undirected graph. A subset I of the vertices is said to be independent if for all $u, v \in I$, $(u, v) \notin E$. A set of vertices M is a *maximal* independent set if M is not contained in any larger independent set, i.e., if $M \subset M'$ then M' is not independent. A set of vertices M is a *maximum* independent set if it is a largest independent set in the graph, i.e., M is independent, and if M' is any other independent set of G, $|M'| \leq |M|$. In other words, a maximal independent set is an independent set that we cannot add any more vertices to without it ceasing to be independent, a maximum independent set is an independent set that contains as many vertices as possible.

a) Show that a maximal independent set is not necessarily a maximum independent set.

b) Show that a maximum independent set is a maximal independent set.

c) Give a polynomial time algorithm that finds a maximal independent set in a graph.

Problem 2 (20 points) Short Answer:

a) How does the run time of finding a minimum cut compare with the run time for finding a maximum flow?

b) What is the run time for the stable marriage algorithm.

c) If you know problem X is NP-complete, and you want to show that problem Y is NP-complete, do you reduce X to Y, or reduce Y to X. Justify your answer.

d) Yes, no, or maybe: is Graph 2-Coloring NP-Complete? Justify your answer.

Problem 3 (15 points) Recurrences:

Give solutions to the following recurrences. Justify your answers.

a)

$$T(n) = \begin{cases} 2T(\frac{n}{2}) + n^3 & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

b)

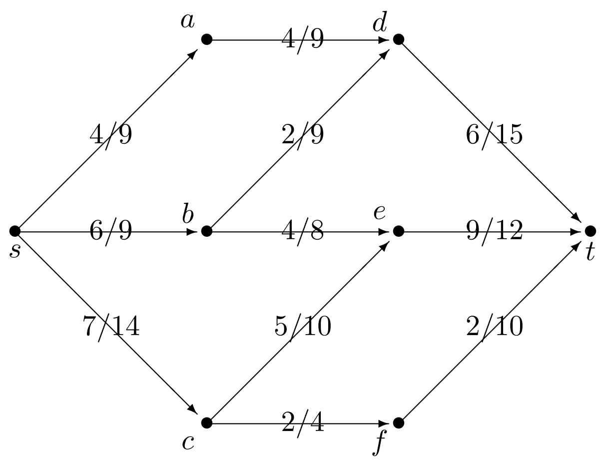
$$T(n) = \begin{cases} 4T(\frac{n}{2}) + n^2 & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

c)

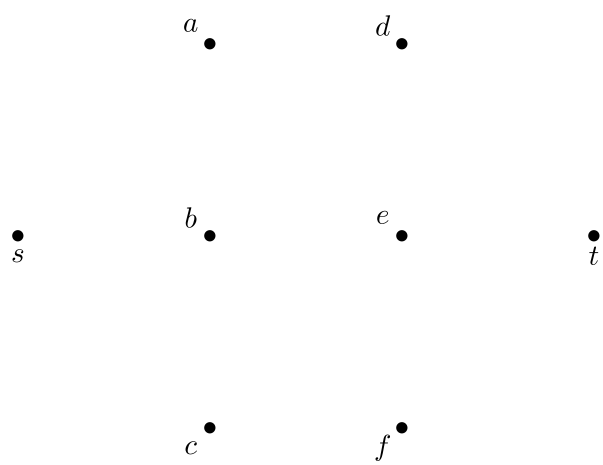
$$T(n) = \begin{cases} 3T(\frac{n}{2}) + n & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

Problem 4 (10 points) Residual Graph:

Consider the following flow graph G, with an assigned flow f. The pair x/y indicates that the edge is carrying a flow x and has capacity y.



a) Draw the residual graph G_f for the flow f.



b) Show how the Ford-Fulkerson algorithm can find a maximum flow. List the paths that you use to augment the flow.

Problem 5 (20 points) How can you?:

Give short answers to the following. You do not need to discuss runtime. How can you:

a) Find a maximum weight spanning tree using an algorithm which computes a minimum weight spanning tree.

b) Find a shortest path in an undirected graph, using an algorithm which finds a shortest path in a directed graph.

c) Determine if there is a path from a vertex s to every other vertex in a directed graph using a network flow algorithm.

d) Find an optimal travelling salesman tour in a graph which may have negative length edges with an algorithm for the travelling salesman problem that requires all edges have positive length.

Problem 6 (15 points) Product-sum:

Given a list of n integers, $v_1, \ldots, v_n$, the *product-sum* is the largest sum that can be formed by multiplying adjacent elements in the list. Each element can be matched with at most one of its neighbors. For example, given the list $1, 2, 3, 1$ the product sum is $8 = 1 + (2 \times 3) + 1$, and given the list $2, 2, 1, 3, 2, 1, 2, 2, 1, 2$ the product sum is $18 = (2 \times 2) + 1 + (3 \times 2) + 1 + (2 \times 2) + (1 \times 2)$.

a) Compute the product-sum of $1, 4, 3, 2, 3, 4, 2$.

b) Give the optimization formula for computing the product-sum of the first j elements.

c) Give a dynamic program for computing the value of the product sum of a list of integers.

Problem 7 (15 points) Electoral college:

The problem is to determine the set of states with the smallest total population that can provide the votes to win the electoral college. Formally, the problem is:

Let p_i be the population of state i, and v_i the number of electoral votes for state i. All electoral votes of a state go to a single candidate, so the winning candidate is the one who receives at least V electoral votes, where $V = \lfloor (\sum_i v_i)/2 \rfloor + 1$. Our goal is to find a set of states S that minimizes the value of $\sum_{i \in S} p_i$ subject to the constraint that $\sum_{i \in S} v_i \geq V$.

a) The dynamic programming solution for this problem involves computing a function OPT where $OPT[i, v]$ gives the minimum populations of a set of states from $1, 2, \ldots, i$ such that their votes sum to exactly v. Give a recursive definition of OPT and an explanation as to why it is correct.

b) What are the base cases for your function OPT.

c) Given an algorithm for computing the smallest population of states that can provide the votes to win the electoral college. (Note that you do not need to find the states).

Problem 8 (15 points) Capacity Reduction for NetFlow:

Let $G = (V, E)$ be a flow graph with maximum flow f. Let e be an edge with capacity c. Suppose that the edge has it's capacity reduced to $c-1$. Describe an $O(n+m)$ time algorithm that computes a new maximum flow f', starting from the flow f for the modified flow graph. Justify the correctness of your algorithm.

Problem 9 (15 points) Currency Conversion:

A group of traders are leaving Afghanistan, and need to convert their Afghanis (the local currency) into various international currencies. There are n traders and m currencies. Trader i has T_i Afghanis to convert. The bank has B_j Afghanis worth of currency j. Trader i is willing to trade as much C_{ij} of his Afghanis for currency j. (For example, a trader with 1000 Afghanis might be willing to convert up to 700 of his Afghanis for USD, up to 500 of his Afghanis for Japaneses Yen, and up to 500 of his Afghanis for Euros).

Assuming that all traders give their requests to the bank at the same time, describe an algorithm that the bank can use to satisfy the requests (if it can).

Problem 10 (10 points) 4SAT:

Show that the 4SAT problem is NP-complete by giving a reduction from 3SAT. 4SAT is the satisfiability problem with exactly four literals per clause. You do not need to show this problem is in NP.