Name: __________________________

CSE 421
Final Exam
March 13, 2007

Instructions:

• You have 1 hour and 50 minutes to complete the exam.

• Feel free to ask for clarification if something is unclear.

• Please do not turn the page until you are instructed to do so.

• The exam is open book, open notes.

• Good luck!
1. (14 points, 2 points for each correct answer, -2 points for each incorrect answer) Indicate for each of the following if it is true or false by circling the appropriate answer.

- **True or False:** You are given a vector \((a_1, a_2, \ldots, a_n)\). You are also given the recurrence \(f(i) = \min_{1 \leq j < i} (j + a_j + f(j))\), and you are told \(f(1) = 0\). Then dynamic programming allows you to calculate \(f(n)\) in time \(O(n)\).

- **True or False:** Consider the problem of shortest paths in a graph \(G\) where edges can have negative weights. Recall that we defined \(Opt(i, v)\) to be the length of the shortest path from \(v\) to \(t\) that uses at most \(i\) edges. Suppose that there is a some vertex \(w\) such that \(Opt(n, w) \neq Opt(n - 1, w)\) (where \(n\) is the number of nodes in the graph). Then \(G\) has a negative cycle.

- **True or False:** Same setup as previous question. Suppose that \(Opt(1, v) = Opt(2, v)\) for all \(v\) (in a graph where there are \(n > 2\) vertices). Then \(G\) has no negative cycle from which \(t\) can be reached.

- **True or False:** We know of a problem in \(NP\) that is also in \(P\).

- **True or False** Suppose that \(X\) and \(Y\) are both in \(P\). Then there is a polytime reduction from \(X\) to \(Y\).

- **True or False:** Suppose that \(X \leq_P Y\), \(X\) is NP-complete and \(Y \in NP\). Then \(Y \leq_P X\).

- **True or False:** Suppose that \(X\) and \(Y\) are both in \(NP\), and that \(SAT \leq_P X\) and \(SAT \leq_P Y\). Then \(X \leq_P Y\).
2. (6 points, 2 points for each correct answer, -2 points for each incorrect answer) Indicate for each of the following if it is **true or false** by circling the appropriate answer. In all of the following, you are given an s-t flow network \( G \), where \( c(u,v) \) is the capacity of edge \((u,v)\).

- **True or False:** Let \( f \) be a maximum flow in \( G \), where \( f(u,v) \) is the flow on edge \((u,v)\). Let \((A_1,B_1)\) and \((A_2,B_2)\) be two different minimum s-t cuts. Then \( \sum_{u \in A_1, v \in B_1} f(u,v) = \sum_{u \in A_2, v \in B_2} c(u,v) \). (Pay close attention to the subscripts.)

- **True or False:** Let \( f \) be a maximum flow in \( G \). Let \((A_1,B_1)\) and \((A_2,B_2)\) be two different minimum s-t cuts. Then \( \sum_{u \in B_1, v \in A_1} f(u,v) = \sum_{u \in B_2, v \in A_2} f(u,v) \). (Pay close attention to the subscripts.)

- **True or False:** Let \( f \) be a maximum flow in \( G \). Let \((A_1,B_1)\) and \((A_2,B_2)\) be two different minimum s-t cuts. Then \( \sum_{u \in A_1, v \in B_1} f(u,v) - \sum_{u \in A_1, v \in B_1} c(u,v) = \sum_{u \in B_2, v \in A_2} f(u,v) \). (Pay close attention to the subscripts.)
3. (12 points, 4 points each) Consider a recursive divide and conquer algorithm that satisfies the following recurrence on its running time:

\[ T(n) = 4T(n/3) + n, \]

with \( T(1) = 1 \). In the following you may assume that \( n \) is a power of 3.

- How many subproblems are there at depth \( k \) in the recursion tree? (The number of subproblems at depth 0, namely at the root of the tree, is 1.)
- What is the size of each subproblem at depth \( k \) of the recursion tree? (The size of the subproblem at depth 0, namely at the root of the tree, is \( n \)).
- What is the running time \( T(n) \) of this algorithm. Express it in big Oh notation as \( O(n^a) \) for an appropriate choice of the value \( a \).

4. (15 points, 5 points each) Recall the knapsack problem: We are given \( n \) items and a “knapsack”. The knapsack can carry a weight up to \( W \). (Assume \( W \) is an integer.) Each item \( i \) has an integer value \( v_i \) and an integer weight \( 0 < w_i < W \). The goal is to choose a subset \( S \) of the items to fill the knapsack with so that \( \sum_{i \in S} w_i \leq W \) and \( \sum_{i \in S} v_i \) is maximized. Let \( V^* \) be the optimum value, i.e. \( V^* = \sum_{i \in S^*} v_i \) for the optimal subset \( S^* \).

In Section 6.4, one dynamic programming approach to this problem is given. Here we explore another.

Define \( OPT(i, v) \) to be the weight of the minimum weight subset of items \( 1 \ldots i \) that yields a total value of exactly \( v \). We get a subproblem for each \( 0 \leq i \leq n \) and each integer \( v \) such that \( 0 \leq v \leq V = \sum_{1 \leq i \leq n} v_i \). (You can define \( OPT(i, v) = \infty \) if there is no subset of \( 1 \ldots i \) that yields value exactly \( v \).)

- Write a recurrence for \( OPT(i, v) \). Be sure to also specify the base cases (\( OPT(0, v) \) and \( OPT(i, 0) \)).
- Given the values of \( OPT(i, v) \) for \( 0 \leq i \leq n \) and \( 0 \leq v \leq V = \sum_{1 \leq i \leq n} v_i \), how would you compute \( V^* \)?
- What is the running time of this dynamic programming procedure for computing \( V^* \)? Include the time to compute the values of \( OPT(i, v) \) for all \( 0 \leq i \leq n \) and \( 0 \leq v \leq V \). Your answer should be expressed in terms of \( n \) and \( V = \sum_{1 \leq i \leq n} v_i \). No need to give the algorithm, just the running time.

5. (15 points, 5 points each) Consider the following network, with edge capacities as shown.
(a) What is the value of the maximum flow in the network? I’m just interested in the single number $\nu(f^*)$.

(b) Give a minimum cut in this network. (Specify which nodes are on the $s$ side of the cut.)

(c) Is there more than one minimum cut in this network? If so, specify which nodes are on the $s$ side of a minimum cut different from the one you gave in part (b).

6. (13 points) Prove that the following problem, called MAXSAT, is NP-complete. Given a 3-CNF formula $\Phi$ and an integer $g$, is there a truth assignment that satisfies at least $g$ clauses?