CSE 101: Practice Final

June 8, 2005

You are allowed a double-sided 8.5x11 page of handwritten notes. Please turn it in along with your test. Use the back pages of the test as scratch paper. If you need additional scratch paper, ask a proctor.

You may not use a calculator of any kind.

If you need to make any assumptions to solve a problem, *write them down* on the test so that we know what you're thinking.

If you are asked to give an efficient algorithm you should:

- Give one that uses the least amount of time (and the least amount of space) as possible
- Show correctness
- Provide a runtime analysis

`Name:` _____

`Student ID:`_____

Problem	Score
1	/13
2	/10
3	/15
4	/15
5	/20
6	/20
7	/20
8	/20
Extra Credit	/20
Total	/133

1. *13 pts.* Choose one of True or False:

True/False Backtracking saves time by not considering sub-trees that can't lead to a valid solution.

True/False Branch and bound removes branches from a decision tree that can't lead to a better solution than has been found so far.

True/False If a polynomial-time algorithm for an NP-complete problem is found, then all problems in NP have polynomial-time algorithms.

True/False NP-complete problems are generally considered to be *hard* problems.

True/False If an augmenting path can't be found in a flow network with flow f, then f is a maximum flow.

True/False For a flow network, the value of the max flow equals the capacity of the minimum cut.

True/False For a flow network with flow f, the value of f equals the net flow of all cuts.

True/False The Floyd-Fulkerson algorithm calculates flows that are integers if the capacities of each edge are integers.

True/False If $f = O(g * h)$, then $f = O(g)$ and $f = O(h)$.

True/False $f = O(g + h)$ if and only if $f = O(min(g,h))$.

True/False If $f = O(g)$ and $g = O(h)$, then $f = O(h)$.

True/False The Floyd-Fulkerson algorithm is a pseudo-polynomial-time algorithm.

True/False The asymptotic complexity of $T(n) = 2T(n/4) + n^{.6} = \Theta(n^{.5})$.

2. *10 pts.* Answer the following

   (a) 2^{60} is approximately what power of 10?

   (b) What is the value of $\sum_{i=0}^{n} i$?.

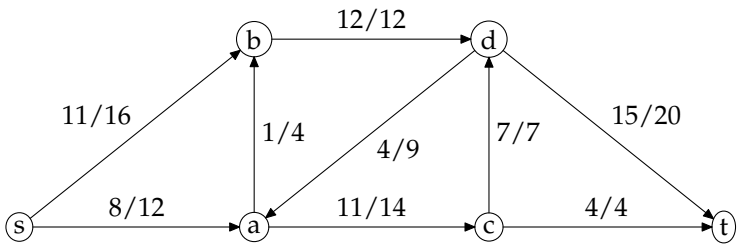
   (c) What is the asymptotic complexity of $T(n) = 9T(n/3) + n^2$?

   (d) How many trees are there in an acyclic graph with n vertices and k edges ($k < n$)?

   (e) If there are n positive numbers $i_1 ... i_n$, and at least half of them are $n/3$ or greater,, what is a lower bound on their total sum?

3. *15 pts.* Show that $T(n) = 3T(n/5) + T(n/5) + n = O(n)$

4. *15 pts.* You're given the following flow network with a current flow, f (each edge is labeled with its flow followed by its capacity) .



Use the Floyd-Fulkerson algorithm to find the maximum flow and the minimum cut starting with the existing flow, f. Show your work.

5. *20 pts.* You are given a list of k undergrad CAPE runners, each of whom provides a list of days on which they can work (CAPE runner i provides a list L_i of days they can work). Each day $1...n$ needs p_i CAPE runners per day. Given $L_1, ..., L_k, p_1, ..., p_n$, find whether a work assignment for each CAPE runner can be made that satisfies the constraints (CAPE runner only works days they're available; all days have enough CAPE runners). If not, report the fact. If so, return the work assignments.

6. *20 pts.* You're given a graph G which is a line of n vertices where each of the $n-1$ edges connects a vertex v_i to vertex v_{i+1} (that is, there are edges $(v_1, v_2), (v_2, v_3), ..., (v_{n-1}, v_n)$). Each of the vertices has a weight w_i. Give an efficient algorithm that returns an independent subset, S, of vertices (no edges between any vertices in S) of maximum total weight.

7. *20 pts.* Consider a buffet that charges a fixed price for a plate filled with as much food as you want, with the restriction that the plate of food is weighed by the cashier and can't exceed a certain weight W. There are n different foods. For each one, you can choose an amount of food between 0 and an entire serving (you can choose $1/2$ a serving, for example, or $1/4$, or a full serving, but not two servings). The weight of a serving of food i is w_i. The number of calories in a serving of food i is c_i.

   Joe is on the crew team and wants to maximize his caloric intake. Give an algorithm to determine the maximum number of calories Joe can get on his plate, subject to the restrictions that:

   - The total weight of the food can't exceed W.
   - The maximum weight of any food i is w_i.

8. *20 pts.* Following is an algorithm that (slowly, but correctly) multiplies two numbers.

```
Procedure Product(i, j)
Precondition: _____________________________________________
_____________________________________________________________
_____________________________________________________________
_____________________________________________________________
Begin
    total = 0
    i2 = i;
    Loop
        Loop Invariant: _____________________________________
        _____________________________________________________
        _____________________________________________________
        _____________________________________________________
        Exit when i2 = 0
        total =total + j
        i2 = i2 - 1
    Endloop
End
Postcondition: total is equal to i * j
```

(a) Fill in the Precondition and Loop Invariant necessary to prove the Postcondition.

(b) Prove the algorithm is true by proving the Loop Invariant and Postcondition. Note: The only information you may use about the loop when proving the Postcondition is the Loop Invariant and the Exit Condition.

9. *Extra credit: 20 pts.*

   Consider the directed Hamiltonian path problem: given a directed graph $G = (V, E)$, is there a path that visits every vertex exactly once?

   Prove that the problem is NP-complete.

   You may find the *Directed Hamiltonian cycle* problem (which is NP-complete) useful: Given a graph G, is there a directed cycle that visits all vertices exactly once?