GT USERNAME:	NAME:

CSE6140 - MIDTERM

Instructions: PLEASE WRITE YOUR NAME and GT ACCOUNT ON EACH PAGE ***VERY CLEARLY IN BLOCK LETTERS***. NUMBER EACH PAGE.

This is a closed-book exam. However, you are **allowed to use one sheet of notes** (both sides of a letter-sized sheet of paper) during the exam. No collaboration is permitted.

You have 80 minutes to complete this exam.

The exam has 10 + 10 + 12 + 12 = 44 points in total.

THIS EXAM HAS **FOUR QUESTIONS**. It is a good idea to read all questions before starting.

1 SHORT QUESTIONS (10 pts)

(a) (2 pts) For each of the following recurrences give the time and space that would be required by a simple dynamic programming algorithm to compute the answer where the values in w are given as input.

Compute OPT(n) where OPT(1) = 1 and $OPT(i) = min_{1 \leq j < i}\{OPT(j)/j + w(j)\}$.
Time Θ(   ) Space Θ(   )

b) (2 pt) Solve $T(n) = 3T(n/3) + \frac{1}{2}n^2 + n$ using the Master Theorem

c) (1 pt) Give a definition of $f(n) \in \Omega(g(n))$

GT USERNAME:	NAME:

(d) (1.5 pts) If problem B is NP-hard and $A \leq_P B$ (A can be reduced in polynomial time to B), then A is NP-hard. **True** **False**

(e) (1.5 pts) If problem A can be solved in $O(n^2)$ and A can be reduced to B in linear time. Then B can be solved in $O(n^2)$. **True** **False**

f) (2 pts) For each pair of functions f and g, choose one of $f \in O(g), f \in \Theta(g), f \in \Omega(g)$ that best describes their relative asymptotic growth. No justification is necessary.

$f = n\log(n)$, $g = \log(n^2)$

$f = 1000n^2 + n$, $g = n^3 + 1000000000$

$f = \sqrt{n}$, $g = \log(n)$

$f = 2^n$, $g = 3^n$

GT USERNAME:	NAME:

2 Greedy: Scheduling with Preferred Finish Times [10 pts]

Here we have a scheduling problem slightly different from the others we have seen. Each job has a length l_i and a preferred finish time f_i, and we must schedule all the jobs on a single machine without overlapping. (There is a single start time $t = 0$ at which all the jobs become available at once.) If we complete a job at completion time c_i before the preferred time, we get a reward equal to the time we have saved. If we complete it after the preferred time, we pay a penalty equal to the amount of time that we are late. Our total reward (which we want to maximize) is the sum of all rewards for early completion minus the sum of all penalties for late completion.

Consider the greedy algorithm where we sort the jobs by length and schedule them in that order, shortest job first, with no idle time. Prove that this achieves a net reward at least as great as that of any schedule (using an exchange argument might be a good idea). You can use the fact that there is an optimal schedule that has no idle time without proving it. Consider also using pictures to illustrate key steps of the argument.

Please define any terms or variables (notation) you use that was not given in the problem.

(Note: In lecture we proved that a different greedy algorithm is optimal for a different goal, that of minimizing the maximum lateness. You cannot simply quote that result because it does not apply to this algorithm or to these rewards and penalties. But a similar *exchange argument* will work in this new case.)

GT USERNAME:	NAME:

.

GT USERNAME:	NAME:

3 DP: Elections [12 pts (+ 2 bonus)]

The problem is to determine the set of states with the smallest total population that can provide the votes to win the election.

Formally, the problem is: We are given a list of states $\{1, ..., n\}$ where each state i has population p_i and v_i, which is the number of electoral votes for state i. All electoral votes of a state go to a single candidate, and the overall winning candidate is the one who receives at least V electoral votes, where $V = (\sum_i v_i)/2 + 1$. 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 $\sum_{i \in S} v_i \geq V$.

The dynamic programming solution for this problem involves computing a function $MinPop(i, v)$ where $MinPop(i, v)$ gives the minimum population of a subset of states from $\{1, 2, ..., i\}$ such that their votes sum to exactly v.

a) (3 pts) Prove optimal substructure for this problem

b) (3 pts) Write the recurrence (include base cases)

c) (4 pts) Give top-down memoization approach . (Hint: We will need to compute $MinPop(n, \sum_{i \in S} v_i)$ and then scan the memo to find the minimum weight of achieving some value $v \geq V$.)

d) (2 pts) What are the time and space requirements in terms of n

e) (2 pts) BONUS: Give bottom-up approach

GT USERNAME:	NAME:

.

GT USERNAME:	NAME:

.

GT USERNAME:	NAME:

.

4 NP-complete (12 pts)

a) (2 pts) Give a formal definition for problem class NP-hard. And draw a diagram of the relationship between classes P, NP, NP-complete and NP-hard (in terms of set inclusion).

b) (10 pts)

LONGEST-CYCLE: Given an undirected graph $G(V, E)$, what is the length of the longest simple cycle in G? The length of a cycle is the number of edges in the cycle. (Remember that a simple cycle is a sequence of vertices starting and ending at the same vertex with no repeted vertices, and where any 2 consecutive vertices in the sequence are connected by an edge.)

Show that the decision version of LONGEST-CYCLE is NP-complete by using the fact the HAMILTONIAN-CYCLE is NP-complete.

Remember in HAMILTONIAN-CYCLE, we are given a graph G, and need to decide whether there is a simple cycle that visits all nodes in the graph.

GT USERNAME:	NAME:

.

GT USERNAME:	NAME:

.