University of New Mexico
Department of Computer Science

Final Examination

CS 561 Data Structures and Algorithms
Fall, 2014

Name:	
Email:	

- This exam lasts 2 hours. It is closed book and closed notes with no electronic devices. However, you are allowed 2 pages of cheat sheets.
- *Show your work!* You will not get full credit if we cannot figure out how you arrived at your answer.
- Write your solution in the space provided for the corresponding problem.
- If any question is unclear, ask for clarification.

Question	Points	Score	Grader
1	20		
2	20		
3	20		
4	20		
5	20		
Total	100		

1. **Short Answer**

   Answer the following questions using *simplest possible* θ notation. Draw a box around your final answer. Briefly justify your answer where appropriate.

   (a) Solution to the recurrence $T(n) = T(n-1) + n^2$

   (b) Solution to the recurrence $T(n) = 2T(n/2) + n^2$

   (c) Solution to the recurrence: $f(n) = 4f(n-1) - 3f(n-2)$.

   (d) Each person in a room chooses an integer uniformly and independently from 1 and n. How many people need to be in the room before the expected number of pairs of people who choose the same number is at least 1?

   (e) In one room, n people write down integers independently and uniformly at random between 1 and n on n slips of red paper. In another room, another n people write down integers between 1 and n on n slips of blue paper, but these people collude and so their numbers are not necessarily uniform and independent. What is the expected number of pairs of red and blue slips that have the same number on them?

(f) Runtime of fastest algorithm to solve the fractional knapsack problem over n items if each item has a value 1, but a weight that is arbitrary?

(g) Number of constraints (i.e. inequalities and equalities) in a linear program for Max Flow over a graph with n nodes and m edges.

(h) In a skip list with n items, each item has height exceeding $3 \log n$ with probability at most $1/n^3$. What is an upperbound on the probability that any item has height exceeding $3 \log n$

(i) Professor Moe has implemented an inefficient Union-Find data structure. His data structure has amortized cost for Make-Set, Find-Set and Union that are all $O(\log n)$. What is the runtime of Kruskal's algorithm on a graph with n nodes and m edges if it uses Professor Moe's implementation?

(j) You have computed a max flow f in a network G with n nodes and m edges, and now a single edge is added to G with capacity 2. What is the cost of the most efficient algorithm to find a new max flow for G?

2. **Short Answer**

(a) (10 points) In the SET-COVER problem, you are given a set S, a collection of subsets of S, $S_1, S_2, \ldots S_\ell$ and an integer k. The problem is to return TRUE iff there exists a $S' \subseteq S$, $|S'| = k$, such that for all $1 \leq i \leq \ell$, $S' \cap S_i$ is not empty (i.e. S' "covers" all subsets S_i).

Show that SET-COVER is NP-HARD by a reduction from one of the following problems: 3-SAT, VERTEX-COVER, 3-COLORABLE or CLIQUE.

(b) (10 points) In the problem MAX-3SAT, you are given a boolean formula f that consists of ℓ clauses, each containing 3 distinct literals. You then must assign truth values to the variables in a way that maximizes the total number of clauses satisfied.

For example, the formula may be $F = (a \vee b \vee c) \wedge (\bar{a} \vee \bar{b} \vee \bar{c}) \wedge (a \vee \bar{b} \vee \bar{c})$. Then an assignment that satisfies all 3 clauses is $a = TRUE$, $b = FALSE$ and $c = FALSE$.

Consider the following randomized approximation algorithm for MAX-3SAT. Assign each variable independently the value TRUE with probability 1/2 and the value FALSE with probability 1/2.

What is the expected number of clauses that this algorithm will satisfy?

3. **Divide and Conquer**

(a) (10 points) Prove by induction that the number of leaves in a binary tree of height h is at most 2^h. Don't forget to include the base case, inductive hypothesis and inductive step.

(b) (10 points) You are given n types of coins with values $v_1, \ldots, v_n$ and a target value x. Assume that $v_1 = 1$ so that it is always possible to make any target value. Give an algorithm to find the smallest number of coins required to sum to x exactly.

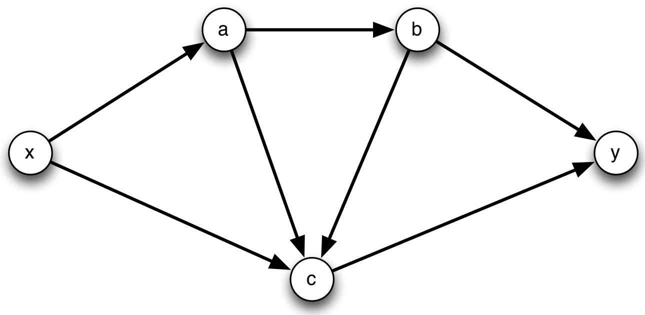
For example, assume the coins have values 1, 6, and 10. Then the smallest number of coins to make 13 is 3: 2 coins of value 6 and 1 of value 1.

Hint: Let $M(i)$ be the minimum number of coins required to make change for the value i. Let S be the set of values $v_1, \ldots, v_n$

4. **Virus**

You are in charge of a computer network, which is modeled as a directed graph $G = (V, E)$. You find out that there is a node x in the network that has become infected with a virus. There is a special node y in the network that you must ensure will never become infected.

For example, the network might be represented as the graph below.



(a) (5 points) Assume that every edge $u \to v \in E$ can be removed at some integer cost $c(u \to v)$. Give an algorithm that will find a subset of edges with minimum cost, whose removal will prevent the infection at node x from reaching node y.

(b) (12 points) Now assume that only nodes can be removed from the network. For each node $v \in V - \{x, y\}$, it is possible to remove v for some integer cost $c(v)$. Give an algorithm that will find a subset of nodes, with minimum cost, whose removal will prevent the infection at node x from reaching node y. Illustrate your algorithm using the figure from part (a).

(c) (3 points) Can you handle the case where the costs for removing nodes are not necessarily integers? If yes, briefly explain how, by using any algorithms discussed in this class. If no, briefly explain why not.

5. **Challenge Problems**

(a) (5 points) You and an opponent are playing a game using a row of n coins of values $v_1 \ldots, v_n$ where n is even. Players take turns selecting either the first or last coin from the row, removing it from the row, and receiving the value of the coin. Assume you play first.

Following are some examples, assuming optimal play for both players:

- 2, 4, 8, 10 : You can collect maximum value 14 (10 + 4)
- 8, 20, 3, 2 : You can collect maximum value 22 (2 + 20)

Give an algorithm that calculates the maximum value you can win assuming that your opponent plays optimally.

(b) (5 points) Now assume that your opponent does not *always* play optimally. In particular, if k coins remain, then they choose the optimal move with probability p_k (for example p_k may decrease as k grows). Describe a new algorithm to optimize your expected winnings.

(c) (5 points) In the Unique Set Cover problem, you are given a set of n elements and a collection of m subsets of the elements. The goal is to pick out a number of subsets so as to maximize the number of uniquely covered elements i.e. those that are contained in exactly one of the picked subsets.

Consider the following naive algorithm. For some number $0 \leq p \leq 1$, the algorithm picks each subset independently with probability p. Assuming that every element is contained in exactly x subsets (x not necessarily a constant), compute the expected number of elements uniquely covered. For what value of p is this expectation maximized? Show that optimal p gives a constant factor approximation algorithm and solve for the constant.

(d) (5 points) Assuming that each element is contained in at most x subsets and at least x/2 subsets, give a constant factor approximation algorithm to Unique Set Cover. Justify your answer and solve for the best constant factor you can achieve.