University of New Mexico
Department of Computer Science

Final Examination

CS 561 Data Structures and Algorithms
Fall, 2013

Name:	
Email:	

- This exam lasts 2 hours. It is closed book and closed notes wing no electronic devices. However, you are allowed a 1 page cheat sheet.
- *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. No need to justify answers for problems on this page.

   (a) $\binom{n}{3}\frac{1}{n^2}$

   (b) *Worst case* runtime of randomized quicksort on a list of n elements?

   (c) Expected number of items at the $\log n$ level of a skip list?

   (d) Amount of space required by a count min sketch used on a data stream containing m items?

   (e) Solution to the following recurrence $T(n) = 2T(n/4) + \sqrt{n}$

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

(g) The time to determine if a weighted graph with n nodes and m edges has a negative cycle that is reachable from a given node.

(h) Recall that in class we showed how to create a Dynamic table where the amortized costs for Insert and Delete were $\theta(1)$. If an algorithm makes $\theta(n)$ calls to Insert or Delete in a table, what is the worst case cost of all of these calls?

(i) What is the worst case cost of a single one of the n calls in the problem above?

(j) Recall that Kruskal's algorithm uses the Union-Find data structure as follows: there are n calls to Make-Set, at most $2m$ calls to Find-Set and at most n calls to Union. In class, we showed that the amortized cost of each of these three operations is $O(log^* n)$ when there are n elements in the sets. Based on these facts, what is the amount of time Kruskals spends on Union-Find operations in the worst case?

(k) You have computed a max flow f in a network G with n nodes and m edges, and now an edge of G has its capacity increase by exactly 1. What is the cost of the most efficient algorithm to find a new max flow for G?

2. **Short Answer**

(a) (10 points) Before a party, n people check their hats. The hats are mixed up during the party so that at the end of the party, each person gets a random hat. In particular, each person gets their own hat with probability $1/n$. What is the expected number of people who receive their own hat?

(b) (10 points) In 4-SAT problem, you are given a boolean formula, f, in conjunctive normal form where each clause has exactly 4 variables, and you are asked if this formula can be satisfied. For example, given $f = (a \vee b \vee c \vee d) \wedge (\neg a \vee \neg b \vee \neg c \vee \neg d) \wedge (a \vee \neg b \vee c \vee \neg d)$, you should return YES since f can be satisfied (for example when a and b are TRUE and c and d are FALSE). Show that 4-SAT is NP-HARD by a reduction from one of the following problems: SAT, 3-SAT, CLIQUE or INDEPENDENT-SET.

3. **Dynamic Programming**

You are given an input string and a dictionary of words, and need to determine if the input string can be segmented into a space-separated sequence of dictionary words. For example, given the dictionary $\{algorithms, data, structure, i, love, snow\}$ and the input string "$ilovealgorithms$", you should output TRUE since the input can be segmented as "i love algorithms".

Assume you have a function "InDictionary(x)" that returns TRUE iff a string x is in the dictionary, and this function runs in $O(1)$ time. As input, you are given a string s, which is represented as an array of length n, i.e. $s = s[1, \ldots, n]$. Define a function f such that $f(i)$ is TRUE iff the substring $s[1..i]$ can be segmented for $0 \leq i \leq n$. Define $s[0]$ to be the empty string.

(a) (15 points) Write a recurrence relation for f.

(b) (5 points) Describe in 1-3 sentences (no need for pseudo-code) how you would create a dynamic program based on your recurrence to find the value of $f(n)$. What are the time and space costs of your algorithm?

4. **Max Flow**

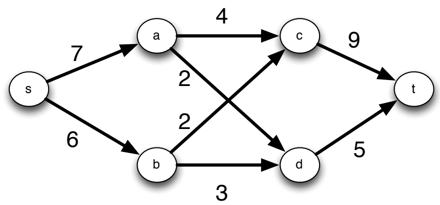

Figure 1

(a) (3 points) Consider the above network (the numbers are edge capacities). Find the max flow, f, and a min cut in this network.

(b) (3 points) Draw the residual graph G_f (along with its edge capacities). In this residual network, mark the vertices reachable from s and the vertices from which t is reachable.

(c) (3 points) An edge of a network is called a *bottleneck* edge if increasing its capacity results in an increase in the maximum flow. List all bottleneck edges in the above network.

(d) (3 points) Give a very simple example (containing at most four nodes) of a network which has no bottleneck edges. All capacities on your network should be finite.

(e) (8 points) Give an efficient algorithm to identify all bottleneck edges in a network. (Hint: Start by running the usual network flow algorithm, and then examine the residual graph.)

5. **Square in Matrices**

You are given a m by n, matrix, M, where each cell is either a "1" or "0". Your goal is to find a maximum size square sub-matrix with all 1's.

```
0 1 1 0 1
1 1 0 1 0
0 1 1 1 0
1 1 1 1 0
1 1 1 1 1
0 0 0 0 0
```

For example, the above matrix has a maximum size square matrix that is 3 by 3, with bottom right corner at $M(5,4)$. Give an efficient algorithm to solve this problem. Compute the time and space costs of your algorithm.

5. **Square in Matrices, continued.**