University of New Mexico
Department of Computer Science

Midterm Examination

CS 561 Data Structures and Algorithms
Fall, 2014

Name:	
Email:	

- This exam lasts 75 minutes. It is closed book and notes, and no electronic devices are permitted. However, you are allowed to use 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 the simplest possible *theta* notation. Assume as usual, that $f(n)$ is $\theta(1)$ for constant values of n.

   (a) Worst case runtime of randomized QuickSort on a list with n elements?

   (b) Expected number of items at the $(1/2)\log n$ level of a skip list containing n items?

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

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

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

2. **Amortized Analysis**

(a) (7 points) Assume we want to use the Union-Find data structure discussed in class in order to detect if there is a cycle in a graph with m edges and n nodes. Let the amortized costs of Union, Find-Set and Make-Set be all $O(\log^* n)$. Briefly describe your cycle finding algorithm and give its worst case cost.

(b) (3 points) What if we have a decimal counter instead of the binary counter discussed in class? The counter starts at all zero digits. The function Increment still adds exactly 1 to the counter and the total time for Increment is the number of digits that are updated. If there are n calls to Increment, what is the worst case cost of a single call?

(c) (10 points) Prove that the amortized cost of Increment is now $O(1)$. Use the accounting method.

3. **Probability and Expectation**

In a k-coloring of a graph, you must assign one of k different colors to each node in the graph. An edge (u, v) in the graph is said to be *satisfied* if u and v have been assigned different colors. In the problems below, assume your graph has n nodes and m edges.

(a) (7 points) Consider the following randomized algorithm for 3 coloring a graph: assign each node a color selected independently and uniformly at random. What is the expected number of edges satisfied by this algorithm?

(b) (7 points) Give an lower bound on the probability that the above algorithm satisfies at least $m/2$ edges. Hint: Let Y be an random variable that is equal to the number of edges that are *not* satisfied.

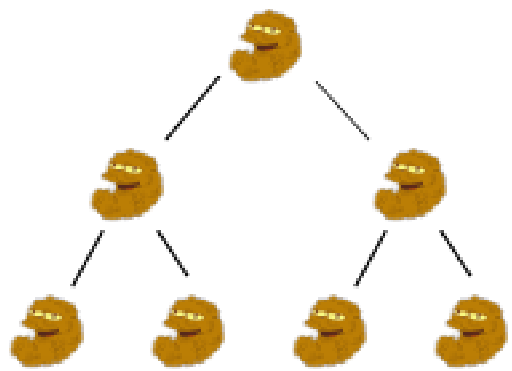
(c) (6 points) Now assume that each node is assigned a color independently and uniformly at random from a set of cm colors for some constant c. What is an upper bound on the probability that some edge is not satisfied? Hint let ξ be the bad event that some edge is not satisfied.

4. **Naming Omicronians**

Omicronians reproduce asexually with each baby producing up to 2 other babies. A birth process results in a binary tree, where nodes are assigned their names in the following way. The node at the root of the tree is assigned the name "J". Then any node that is a left child of a node with name σ receives the name σL and any node that is the right child of that node receives the name σR.

Your goal is to take a binary tree specifying a birth process and return the number of R's in the names of all nodes.

Example: In the following tree, the names are J, JL, JR, JLL, JLR, JRL and JRR, so the total number of R's is 5.



(a) (10 points) Give a recurrence relation for the number of R's in the names of all nodes.
Hint: For a node v, let $f(v)$ be the number of R's in the tree rooted at v, if the naming started at v. For simplicity, if v is NULL, let $f(v) = 0$ and let $n(v) = 0$. Also, for a node v, let $l(v)$ (resp. $r(v)$) be the left (resp. right) child of v if it exists or NULL otherwise. Finally, for a node v, let $n(v)$ be the number of nodes in the subtree rooted at v and assume this value is stored at each node. Now give a recurrence relation for $f(v)$.

(b) (4 points) Your friend Bender claims that once you have the recurrence relation, you can just write a recursive algorithm for this problem in order to compute $f(r)$ where r is the root node. There's no need to bother with a dynamic program that stores the computed f values at the nodes of the tree. Is Bender right? Explain in a couple of sentences.

(c) (6 points) The Omicronians have evolved! Now they can reproduce both sexually and asexually: i.e. each node can have 1 or 2 parents. So a birth process is now a rooted directed graph without cycles. Now each node can have multiple names, one for each path from the root down to that node. A particular path determines a name in the same way as before: start with J at the root and add a L whenever a left edge is taken and a R whenever a right edge is taken.

Professor Farnsworth claims that the same recurrence relation from part (a) can be used for this problem. Bender claims that this problem also can be done recursively, without the dynamic programming approach of storing f values at intermediate nodes. Which if either of your friends are correct? Explain briefly.

5. **Probability**

There are two bins. Bin 1 initially has 3 white balls and 1 red ball. Bin 2 has 4 white balls. In every round, a ball is selected uniformly at random from each bin and these two balls are swapped.

Let p_k be the probability that the red ball is in bin 1 at the beginning of the k-th round.

(a) (8 points) Write a recurrence relation for p_k.

(b) (8 points) Use the guess and check and proof by induction method to solve this recurrence. Don't forget to label BC,IH and IS and clearly say where you are using the IH.

(c) (4 points) Assume you are paid one dollar for every round in which the red ball is in bin 1 and there are m rounds. What is the expected number of dollars you earn?