University of New Mexico
Department of Computer Science

Final Examination

CS 362 Data Structures and Algorithms
Spring, 2008

Name:	
Email:	

- Print your name and email, *neatly* in the space provided above; print your name at the upper right corner of *every* page. Please print legibly.
- This is an *closed book* exam. You are permitted to use *only* two pages of "cheat sheets" that you have brought to the exam and a calculator. *Nothing else is permitted.*
- Do all the problems in this booklet. *Show your work!* You will not get partial credit if we cannot figure out how you arrived at your answer.
- Write your answers in the space provided for the corresponding problem. Let us know if you need more paper.
- Don't spend too much time on any single problem. The questions are weighted equally. If you get stuck, move on to something else and come back later.
- If any question is unclear, ask us for clarification.

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

1. **Short Answer**

   (a) Assume there is some function *foo* that has amortized cost of $O(\log n)$. If *foo* is called n times in a row, what is the maximum cost of any single call to *foo* in this sequence of calls?

   (b) Consider the following problem. You are given a connected graph G with n nodes and m edges. You want to remove as many edges as possible and still keep the graph connected.

   - How many edges must remain in the graph in order to ensure that G is connected?

   - Give an efficient algorithm to solve the problem

(c) Assume you have a sorting algorithm that runs in linear time worst case (obviously this would have to be a non-comparison based sorting algorithm). If you use this new sorting algorithm, what will be the new asymptotic run time of Kruskal's on a graph with n nodes and m edges? Assume the amortized cost of all operations on the union-find data structure is $\log^* n$. Justify your answer.

(d) Assume you are given as input a directed graph without cycles (this is commonly called a directed acyclic graph or DAG for short), and a special source vertex s that has no in edges. All other vertices in the graph have in-edges, and the edges on the graph have both positive and negative weights. How quickly can you find the *longest* path in this graph that starts at s? Justify your answer.
Hint: This is easy if you use an algorithm from class in the correct way.

2. **Recurrences and Asymptotics**

(a) Give an asymptotic solution for the following recurrence: $T(n) = 2T(n/2) + n^2$

(b) Solve the following recurrence $T(n) = 5T(n-1) - 6T(n-2)$. Give the solution in general form i.e. do not solve for the constants.

(c) Consider the recurrence $T(n) = 2^{\log T(n/2)} + T(n/2)$, where $T(1) = 1$

Use induction to prove that $T(n) = O(n)$. Don't forget to write down precisely what you are trying to prove and to include the base case, inductive hypothesis and inductive step.

3. **PARTITION**

In the PARTITION problem, you are given a list of n integers, $x_1, x_2, \ldots x_n$ all in the range 1 to k. You want to determine if there is some subset of the integers that sums to exactly $S/2$, where S is the sum of all the integers in the list.

(a) Give an algorithm to solve this problem that has runtime polynomial in n and k, and analyze your algorithm. Hint: Let $f(i, y) = 1$ if there is some subset of the first i integers that sums exactly to y and let it be 0 otherwise.

(b) The PARTITION problem is NP-Hard. Explain how this fact reconciles with your ability to devise an efficient algorithm above. Hint: This answer is no more than two sentences.

4. **Minimum Spanning Trees**

   Suppose you are given a connected graph $G = (V, E)$ with n nodes and m edges. Professor Bunyan claims that an edge $e = (u, v)$ is in a minimum spanning tree of G if there is no path from u to v in G that consists only of edges that have weight cheaper than e. Is Professor Bunyan correct? If so, prove it. If not, give a counter example.
   Hint: Think about the safe-edge theorem.

5. **Independent Set**

Recall that an Independent Set for a graph $G = (V, E)$ is a set of vertices V' such that no pair of vertices in V' have an edge between them in G. In the problem INDEPENDENT-SET, you are given a graph G, and an integer k and must output TRUE if there is an independent set of size k in G and FALSE otherwise.

(a) Imagine you have a polynomial time algorithm that solves INDEPENDENT-SET. Show that you can use this algorithm to create a polynomial-time algorithm that takes as input a graph G and integer k and actually outputs an independent set of size k if it exists.

Hint: You will need to make multiple calls to the algorithm that solves INDEPENDENT-SET, essentially one for each vertex in G.

Now you will describe an approximation algorithm for the independent set problem.

(b) If the minimum vertex cover of a graph G is of size x, what is the size of the maximum independent set? Assume that G has n nodes and m edges.

(c) Assume the smallest vertex cover of G is less than or equal to $n/4$. Describe a good polynomial time approximation algorithm for finding the maximum independent set of G.

5. **Independent Set, continued.**