University of New Mexico
Department of Computer Science

Midterm Examination

CS 561 Data Structures and Algorithms
Fall, 2007

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 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. **Asymptotic Analysis and Recurrence Relations**

   Is $n/2 \in o(n)$? Prove your answer using definitions of asymptotic notation given in the book and in class and solve for the values required to show the definitions hold or do not hold.

   Recurrence Relations: Consider the Recurrence $T(n) = 4T(n-1) - 4T(n-2) + 3^n$. Write down the general form of the solution for this recurrence (i.e. don't solve for the constants).

2. **Heaps and Sorting**

   Prove that you cannot build a Priority Queue in the comparison model (i.e. only comparison operations $\leq$, $\geq$, $=$, etc. are allowed on the keys) with both of the following properties:

   - Extract-Min runs in $\Theta(1)$ time
   - Build-Heap runs in $\Theta(n)$ time

   Prove *succinctly* that the following algorithm correctly sorts a list of n elements by induction on n. Don't forget to include the base case, inductive hypothesis and inductive step.

```
GoofySort(A,i,j){
  if i+1 > j
        then return;
  Let s be the index of the minimum element in A[i..j];
  Exchange A[1] and A[s];
  Let b be the index of the maximum element in A[i..j];
  Exchange A[j] and A[b];
  GoofySort(A,i+1,j-1);
}
```

3. **Search Trees**

Consider a tree with the following properties:

- Each internal node has exactly three children
- The heights of the subtrees rooted at each child differ by at most 1.

What is the maximum height of such a tree containing n nodes?

Hint: Write a recurrence relation for the maximum number of nodes as a function of the height and then solve for the height. Show your work!

4. **Hash Tables and Probability**

   Assume we hash n items into a hash table with n bins using a good hash function i.e. each item is hashed to a bin chosen independently and uniformly at random. Give a good upper bound on the number of empty bins. Solve for the constants in your upper bound i.e. do not use asymptotic notation.

   Hint: Use the fact that $1 - x \leq e^{-x}$ for all x.

5. **Divide and Conquer**

   Imagine that after graduating from UNM, you start your new job at the exciting investment banking firm SELLOUT, Inc. The firm if faced with the following problem: they have an array of the predicted prices of a stock over n days and they want to determine, using this array, exactly one day to buy the stock and one day to sell the stock in order to maximize their profit.

   The problem can be formally stated as follows. You are given an array A of n numbers. You want to choose indices $1 \leq i < j \leq n$ such that $A[j] - A[i]$ is maximized over all such indices. Give an $o(n^2)$ algorithm to solve this problem.