CS161 - Midterm Exam
Computer Science Department, Stanford University
July 21, 2008

Name: _____

Honor Code

1. The Honor Code is an undertaking of the students, individually and collectively:

   (a) that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;

   (b) that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

2. The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.

3. While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

Signature: _____

1. True/False (20 points) - State whether the statements below are true or false and give a brief justification for your answer.

_____A list of items can be sorted in $\Theta(n)$ time

_____The only way a B-Tree grows in height is by splitting the root

_____In a max-heap, the smallest element has to be one of the leaves

_____$f(n) + g(n) = \Theta(\max(f(n), g(n))$

_____Every call to RandomizedQuicksort takes time $n \log n$

_____A complete binary tree has $n/2 - 1$ leaves

_____If $f(x) = O(g(x))$ then $f(100) > g(100)$

_____$2^{10n} = O(2^n)$

_____$h(x) = \lfloor x/100 \rfloor$ is a valid hash function

_____The time to find a key in a binary search tree is $O(\log n)$

2. (10 points) kth Sorted Value

Suppose you are given an array A containing n unique integers and you are asked to find the value stored at index k if A were first sorted. One way to do it is with the following function:

```
KTH(A, k, p, r)
1  index = PARTITION(A, p, r)
2  if index = k
3          return A[index]
4  elseif index > k
5          return kth(A, k, p, index − 1)
6  else
7          return kth(A, k, index + 1, r)
```

(a) What call would you make to KTH given an array A to find the value of the $8th$ smallest value?

(b) 5th largest value?

(c) What is the best-case running time of KTH on an array of length n? Explain your answer.

(d) What is the worse-case running time of KTH on an array of length n. Explain your answer.

3. (10 points) Short Answer

   (a) Name four properties of algorithms that are important to consider when designing an algorithm.

   (b) Take a linear-probed hash-table of length n where a key k is first probed at $|hash(k)|$ mod n where $hash$ is a hash function. In addition to storing the key and data value, it can also be beneficial to store $hash(k)$. There are two advantages hinted at below. For each, explain the advantage in more detail.

   - To speed up lookup operations

   - To speed up the proces of doubling the size of the table when the load becomes too high

   (c) In what situation does quicksort achieve it's best case running time? Determine an 8 element array such that the quicksort procedure achieves this running time.

4. (20 points) If possible, solve the following recurrences (note, master method found in appendix)

   (a) $T(n) = 6T(n/3) + n^2 \log n$

   (b) $T(n) = T(n+1) + n$

   (c) $T(n) = T(n-1) + n^3$

   (d) $T(n) = 4T(n/2) + n^2$

5. (15 points) Egg Drop

   You are asked to run an experiment to determine how high you can drop an egg before it breaks. You've been given access to a very high building and can drop an egg from any floor of this building. If you drop an egg from too tall a height, the egg will break and you can no longer use that egg. For your algorithms you want to *minimize* the number of times you drop an egg. Points will be deducted for inefficient algorithms.

   (a) Given two eggs, describe an algorithm for determining the tallest height an egg can be dropped without breaking. State the running time, in number of drops, of your algorithm using O or Θ, whichever is more appropriate.

   (b) What about given k eggs?

6. (15 points) Binary Search Trees

(a) Suppose that we have numbers between 1 and 1000 in a binary search tree and want to search for the number 363. Which of the following sequences could not be the sequence of nodes examined?

(a) 2, 252, 401, 398, 330, 344, 397, 363
(b) 924, 220, 911, 244, 898, 258, 362, 363
(c) 925, 202, 911, 240, 912, 245, 363
(d) 2, 399, 387, 219, 266, 382, 381, 278, 363
(e) 935, 278, 347, 621, 299, 392, 358, 363
(f) 1000, 999, 998, 997, ..., 364, 363

(b) Is the operation of deletion in a binary search tree commutative in the sense that deleting x and then y from a binary search tree leaves the same tree as deleting y and then x? Argue why it is so or give a counter-example.

(c) Write a recursive function NumTrees that, given the number of distinct values, computes the number of structurally unique binary search trees that store those values. For example, NumTrees(4) should return 14, since there are 14 structurally unique binary search trees that store 1, 2, 3, and 4.

7. (10 points) Runtime Comparison

   Consider the following information:

$$f_1 = O(n) \quad f_2 = O(n \log n) \quad f_3 = \Omega(n^2)$$
$$f_4 = \Omega(\log n) \quad f_5 = \Theta(1) \quad f_6 = \Theta(n^2)$$

   From this, we can conclude (for sufficiently large input) that $f_5 < f_6$. Give **all** other *pairs* of functions for which you can establish an ordering and specify the ordering.

Appendix

Master Theorem

$T(n) = aT(n/b) + f(n)$

- if $f(n) = O(n^{\log_b a - \epsilon})$ for $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$
- if $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \log n)$
- if $f(n) = \Omega(n^{\log_b a + \epsilon})$ for $\epsilon > 0$ and $af(n/b) \leq cf(n)$ for $c > 1$ then $T(n) = \Theta(f(n))$