CSCI 270 Homework #2

Due Date: Thursday, September 26th, 2pm

Submit in class or the dropbox (Box 3, first floor of SAL, opposite the Men's bathrooms).
Solutions will be posted Friday evening, so you may only use a single late day on this assignment.

1. Write your name, student ID Number, and which lecture you attend (morning or afternoon). Multi-page submissions must be stapled.

2. Aaron is trapped on an island with hungry velociraptors. He is currently at a safe location s, and he wants to get to the boat t which will get him home. There are many paths through the island, represented as a directed graph. Each edge e has a probability p_e that a traveler along this edge will be eaten by a raptor. If we select a path P from s to t, the probability he arrives safely is $p_{safety} = \Pi_{e \in P}(1 - p_e)$. Give an efficient algorithm which maximizes the probability he arrives safely. *Note: Please do not feed your instructor to hungry dinosaurs.*

3. Suppose that I pick an integer x between 1 and n. Your job is to identify x using the fewest number of "yes/no" questions possible. You may select any subset of the numbers between 1 and n, and ask if x is within that subset.

   (a) Explain how to identify x using the smallest number of queries.

   (b) Analyze the number of queries needed for your solution to part a, by setting up and solving a recurrence relation.

   (c) Suppose that I am a particularly sadistic instructor, and that I am allowed to lie exactly once. You can still solve this problem by simply asking each question twice (and a third time when you detect the lie). Analyze the number of queries needed in this strategy.

   (d) Assuming I am still allowed to lie once, explain how we can eliminate $\frac{1}{4}$ of the possible numbers using only two queries.

   (e) Analyze the number of queries needed using your solution to part d, by setting up and solving a recurrence relation. Which solution (part c or part d) is better overall?

4. Suppose we have an array A allocated in memory, but we do not know the size of this array (and there is no member variable or function of A which tells us the size of the array). We can query the contents of $A[i]$ in constant time, but if i is greater than the bounds of the array, we will simply get back the unhelpful error: "index out of bounds." If we want to identify the size of the array, we could query the contents of each index until we find the first error, but this would take $\theta(n)$ time, where n is the (currently unknown) size of the array. Give a more efficient algorithm to determine the size of the array, and analyze the running time in terms of n.

   Continued on Back.

5. Suppose we are given all the elements we wish to insert into an initially empty MinHeap up front. If we insert one at a time, it takes $O(n \log n)$ time to build the MinHeap. Suppose we instead build a "pre-heap" of depth d by forming a complete binary tree with each node in an arbitrary position. We must "heapify" this pre-heap so that each node has value $\leq$ its children.

   (a) Suppose we take the subtrees of size ≤ 3 rooted at depth $d-1$. Show how to heapify **all** of these subtrees in $O(n)$ total time.

   (b) Assuming that all subtrees rooted at depth j are valid MinHeaps, show how to heapify **a single** subtree rooted at depth $j-1$ in $O(d-j)$ total time.

   (c) Analyze the running time to build a heap, using the inductive design process outlined in parts a and b. *You may find the following summation useful:* $\sum_{i=1}^{n} \frac{i}{2^i} = O(1)$.

If you would like some extra practice, you may do the following problems. Do not submit them, as they will not be graded. If you would like to check your answers, talk to the instructor or TA via email or office hours. All extra practice problems are from the Kleinberg and Tardos textbook.

Chapter 5: exercises 1, 2, 6