



Assignment 5
Due Thursday, 10/11

October 8, 2018
CS: DS&A
PROOF SCHOOL

This is a theoretical assignment with only two questions. The first one reasonable; the second one is tricky. If you're stuck, come find me on Wednesday and ask for a hint!

Write up on computer and submit to Classroom, or write up on paper and give to me in person.

1. Suppose you have a complete binary tree where the root is labeled 1, the nodes on the next level are labeled 2 and 3, etc. You have a calculator that has basic calculator functions, including \log_2 , exponentiation, and conversion from decimal to binary.

a) How do you use your calculator to determine the height of a node labeled n ? (The height of the root is 0; the height of the next two nodes is 1, etc.) Your approach should allow you to determine the height of the 10,000th node, for example, in a few seconds with your calculator.

b) Each node can be described by a sequence of 0's and 1's that gives you directions to that node from the root, where 0 means go left, and 1 means go right. For example, node 4 corresponds to "00", while node 14 corresponds to "110". How do you use your calculator to easily compute this sequence for any given node n ?

2. You are given a binary heap with n elements. Try to find an algorithm that tells you whether a given x is greater than or equal to the k th smallest element in the heap, that runs in $O(k)$ time, independent of the size of the heap.

Hint/thought. One approach is to pop the least k elements from the heap and compare them to x , but that takes $O(k \log n)$ time. The trick is that I'm not actually asking you to determine the k th least element in the heap. (You couldn't do that in $O(k)$ time.) I'm just asking you to determine if x is \geq the k th element or not. So how do you determine this without actually knowing the k th element?

I have a hint I can give you if you want it. But you have to think about this first, and you'll have to tell me a joke.