15-451/651 Algorithms, Spring 2019

Homework #Bonus **Due: ~~February 3~~ February 10, 2019**

(1 bonus) B1. **(Where's the Loot?)**

Given an unweighted undirected graph, there is treasure at one of the nodes (but you don't know the identity of that node).You are allowed the following query operation. If you query node x: if x contains the treasure, you find that out. Else you are told the identity of some edge (x, y) incident to the queried node x that lies on a shortest path from x to the treasure node.

Give an algorithm that for any graph finds the treasure in $O(\log n)$ queries. Prove a matching lower bound: that there exist graphs which require at least $\Omega(\log n)$ queries to find the treasure, in the worst case.

(1 bonus) B2. **A Set of Pills.**

There are n red pills and n blue pills. The weights of all pills, blue and red, are distinct. Moreover, if the weights are denoted by $b_1, b_2, \ldots, b_n$ for the blue pills and $r_1, \ldots, r_n$ for the red pills, then you are guaranteed the "interleaving property" that

$$b_1 < r_1 < b_2 < r_2 < \ldots < b_n < r_n.$$

You have a balance scale that allows you to put a red pill on one side and a blue pill on the other side. In one weighing, it tells you if the red one is heavier, or the blue one is heavier. You are not allowed to weigh two pills of the same color.

You want to find the median weight red pill (i.e., the red pill $b_{\lfloor n/2 \rfloor}$) but only using blue-red comparisons. Clearly, if you do all the n^2 comparisons, you can find this pill.

Give a randomized algorithm where the expected number of weightings is $O(n(\log n)^c)$ for some constant c independent of n. Ideally $c = 0$, but you will get the bonus point for any constant $c \leq 4$, say. If you can do this in a deterministic way, that's even better.