Design and Analysis of Algorithms

Problem #1.

Design an $O(\log n)$ time algorithm that determines whether a red-black tree with n keys stores any keys within a certain (closed) interval. That is, the input to the algorithm is a red-black tree T and two keys, l and r ($l \leq r$). If T has at least one key k such that $l \leq k \leq r$, then the algorithm returns *true*, otherwise it returns *false*.

Problem #2.

As your reward for saving the Kingdom of Bigfunnia from the evil “Exponential symptotic" the king has given you the opportunity to earn some “big cash". Behind the castle (near the hot tub) there is a maze. Along each corridor of the maze there is a bag of gold coins. The amount of gold in each bag varies. You will be given the opportunity to walk through the maze, picking up bags of gold. You may enter only through the door marked “ENTER" and exit through the door marked “EXIT". (These are separate doors). While in the maze you may not retrace your steps. Each corridor of the maze has an arrow painted on the wall. You may only go down the corridor in the direction of the arrow. There is no way to traverse a “loop" in the maze. You will receive a map of the maze, including the amount of gold in and the direction of each corridor. Describe an algorithm to help you pick up the most gold?

Problem #3.

You are implementing an algorithm that draws part of the landscape of a terrain, and you are faced with the following problem: You are given the heights of n points of the terrain's grid, and you need to find and sort, as fast as possible, the $\frac{n}{\log n}$ highest of them. Give an $O(n)$ time algorithm that does this.