Design and Analysis of Algorithms

**Problem #1**
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 \(\sqrt{N}\) highest of them. Give an algorithm that does this, and argue that no one can do better (up to a constant factor, of course!) \((Hint: Your algorithm should run in \(O(N)\) time; there is a simple argument why this is the best possible.)\)

**Problem #2**
Professor Marley hypothesizes that he can obtain substantial performance gains in hashing by modifying the chaining scheme to keep each linked list in sorted order. How does the professor’s modification effect the running time for successful searches, unsuccessful searches, insertions, deletions?

**Problem #3**
A directed graph is *singly-connected* if there is at most one simple path from \(u\) to \(v\) for all vertex pairs \(u, v\). Give an efficient algorithm to determine whether or not a directed graph is singly-connected.