MID-TERM

Remarks: Solve 4 of the following 5 questions. In all algorithm, always explain how and why they work. ALWAYS, analyze the complexity of your algorithms. In all algorithms, always try to get the fastest possible. A correct algorithm with slow running time may not get full credit.

1. Question 1:
   (a) Give a non-recursive version of Quicksort that uses a stack. The version should store at the stack indices \([i, j]\) of subarrays that have not been sorted yet.
   (b) What is the maximum number of of \([i, j]\) pairs (non-sorted subarrays) that can belong to the stack simultaneously? Give an example where your bound is achieved.

2. Question 2: In the following questions, when its a true or false question, do not only say true or false. Explain your answer.
   (a) What is the difference between the worse case and best case running time of Mergesort?
   (b) True or false: Let \(A^i\) and \(A^{i+1}\) be the \(i\) largest and the \(i+1\) largest elements in \(A\). Any algorithm for sorting (based on comparisons) must compare \(A^i\) and \(A^{i+1}\).
   (c) What has to happen so that the maximum and minimum in an array will be compared during Quicksort? Assume that the numbers are pairwise different.
   (d) Let \(A\) be an algorithm for a problem that runs in \(O(n^2)\) and \(A'\) an algorithm for the same problem that runs in \(O(n)\). True or false: Algorithm \(A'\) runs better than \(A'\) in practice.
   (e) What is the running time of insertion sort on a sorted array? On an array sorted in reverse?

3. Question 2: You are given two identical crystal balls and an \(n\) stories building. The goal is to find the least floor so that if you throw the ball from this floor, the ball will break. The rules are:
   (a) The two balls break at the same floors.
   (b) If a ball breaks at floor \(i\) it also breaks at floor \(i + 1\)

Give an algorithm to find the least floor the balls breaks at with minimum number of possible ball throws.

4. Question 4: Let \(S_1\) and \(S_2\) be two strings. We say that we extend \(S_1\) if we add a letter at the beginning or at the end of \(S_1\) (but not in the middle).
**Example:** If $S_1 = ababc$. Then both $xababc$ and $ababcx$ extend $S_1$.

Give an algorithm that checks if we can get from $S_1$ to $S_2$ by a sequence of extensions.

5. **Question 5:** Give an algorithm for subset sum for the special case that:

   (a) All $A[i]$ are powers of 2.
   (b) All the numbers in the array are pairwise different