Algorithms I

Tutorial 5

[redacted]

Problem 1

CLRS 16.3-5

Problem 2

A sequence is called a good sequence if $a_1 < a_2 > a_3 < a_4 \dots a_k$. i.e. $a_i < a_{i+1}$ if i is odd and $a_i > a_{i+1}$ if i is even for all $i < k$. You are given a sequence A containing n integers. You need to find the length of longest good subsequence of A in $O(n^2)$ time.

Problem 3

You are given integers n and k, along with $p_1, \dots, p_n \in [0, 1]$, you want to determine the probability of obtaining exactly k heads when n biased coins are tossed independently at random, where p_i is the probability that the i^{th} coin comes up heads. Your algorithm should run in $O(n^2)$ time.

Problem 4

You are given n intervals $(l_1, r_1), (l_2, r_2), \dots (l_n, r_n)$, l_i and r_i are integers and $l_i \le r_i$ for all $1 \le i \le n$. You need to find a set of non-overlapping intervals such the sum of the lengths of the intervals in the set is maximized. Your algorithm should run in $O(n^2)$ time. Can you improve it to $O(n \log n)$?

Problem 5

You are given a set A of n positive integers $a_1, a_2, \dots a_n$ and a positive integer k. You need to check whether there exists a subset S of original set of integers such that the sum of all the elements in S is a multiple of k in $O(nk)$ time.

Problem 6

You are given a string S of length n. Your task is to find the length of the longest subsequence of S which is a palindrome in $O(n^2)$ time.

Problem 7

Alice and Bob are playing a two player game. There is a pile having n stones. In each turn a player can remove at most k stones and he/she must remove at least one stone. The player who is unable to remove a stone loses. Alice starts first. Your task is to find whether Alice an guarantee her win if both players play optimally. Here, optimally means if a player can ensure his/her win, he/she will. Your algorithm should run in $O(nk)$ time.

Problem 8

You are given a $n \times m$ grid. Each cell (i, j) in the grid has a_{ij} coins. If you are at a cell (i, j), you can only move to $(i + 1, j)$ or $(i, j + 1)$ given the next cell lies inside the grid. You start from $(1, 1)$ and your task is to reach (n, m) with maximum number of coins. You need to find the maximum number of coins you can end up with in $O(nm)$ time.