Homework #4
Introduction to Algorithms/Algorithms 1
601.433/633
Spring 2020

Due on: Tuesday, March 10th, 12pm

Where to submit: On Gradescope.
Please type your answers; handwritten assignments will not be accepted.

1 Problem 1 (15 points)

Suppose you are managing the construction of billboards on an east-west highway that extends in a straight line. The possible sites for billboards are given by reals $x_1, x_2, \ldots, x_n$ with $0 \leq x_1 < x_2 < \cdots < x_n$, specifying their distance in miles from the west end of the highway. If you place a billboard at location x_i, you receive payment $p_i > 0$.

Regulations imposed by the Baltimore County's Highway Department require that any pair of billboards be more than 5 miles apart. You'd like to place billboards at a subset of the sites so as to maximize your total revenue, subject to that placement restriction.

For example, suppose n = 4, with

$$\langle x_1, x_2, x_3, x_4 \rangle = \langle 6, 7, 12, 14 \rangle,$$

and

$$\langle p_1, p_2, p_3, p_4 \rangle = \langle 5, 6, 5, 1 \rangle.$$

The optimal solution would be to place billboards at x_1 and x_3, for a total revenue of $p_1 + p_3 = \$10$.

Give an $O(n)$ time dynamic-programming algorithm that takes as input an instance (locations $\{x_i\}$ given in sorted order and their prices $\{p_i\}$) and returns the maximum revenue obtainable. As usual, prove correctness and running time of your algorithm.

EDIT: We will give full credit for an $O(n \log(n))$ algorithm too.

2 Problem 2 (20 points)

You are given two numbers, n and k, such that $n \in \mathbb{N}$ and $k \in \{1, \ldots, 9\}$. Use dynamic programming to devise an algorithm which will find the number of $2n$-digit integers for which the sum of the first n digits is equal to the sum of the last n digits and each digit takes a value from 0 to k.

For example, when $k = 2$ and $n = 1$: you have only 3 such numbers 00, 11, 22. For example, when $k = 1$ and $n = 2$: you have only 6 such numbers 0000, 0101, 0110, 1001 , 1010, 1111.

Your algorithm should work in time polynomial of n and k. Prove correctness and provide running time analysis.

3 Problem 3 (15 points)

Alice and Bob found a treasure chest with different golden coins, jewelry and various old and expensive goods. After evaluating the price of each object they created a list $P = \{p_1, \ldots, p_n\}$ for all n objects, where $p_i \in \{1, \ldots, K\}$ is the price of the object i. Help Alice and Bob to check if the treasure can be divided equally, i.e. if it is possible to break the set of all abjects P into two parts P_A and P_B such that $P_A \cup P_B = P$, $P_A \cap P_B = \emptyset$ and $\sum_{i \in P_A} p_i = \sum_{i \in P_B} p_i$?

Your algorithm should run in time polynomial in n and K. As usual, prove correctness and running time of your algorithm.