Homework #4
Introduction to Algorithms/Algorithms 1
600.363/463
Spring 2017

Due on: Tuesday, March 7th, 11:59pm
Late submissions: will NOT be accepted
Format: Please start each problem on a new page.
Where to submit: On Gradescope, under HW4
Please type your answers; handwritten assignments will not be accepted.
To get full credit, your answers must be explained clearly,
with enough details and rigorous proofs.

February 26, 2017

1 Problem 1 (20 points)

When you are checking out at BalMart, you want to make change for A cents. Assuming that the cashier has infinite supply of each of $C = \{C_1, C_2, \ldots, C_t\}$ valued coins, can you count how many ways to make change for A cents? Give an efficient dynamic programming algorithm and analyze the running time. Here we don't consider the order of the coins.

For example, when $A = 3$ and $C = \{1, 2, 3\}$, there are three solutions: $\{1, 1, 1\}$, $\{1, 2\}$, $\{3\}$.

2 Problem 2 (20 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 numbers $x_1, x_2, \ldots, x_n$ 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

$$< x_1, x_2, x_3, x_4 >=< 6, 7, 12, 14 >,$$

and

$$< p_1, p_2, p_3, p_4 >=< 5, 6, 5, 1 > .$$

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 efficient 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 from a valid subset of sites. Analyze the running time of your algorithm. Your solution must clearly define a recursive formula.