

CS3100: Algorithms

Dynamic Programming

In class, Sept. 18

Problems

1. Advertising on the Highway

Suppose you are managing the construction of billboards on the Information Highway, a heavily traveled road that runs west-east for M miles. The possible sites for billboards are given by numbers x_1, x_2, \dots, x_n , each in the interval $[0, M]$ (specifying their position along the highway, measured in miles from its western end). If you place a billboard at location x_i , you receive a revenue of $r_i > 0$.

Regulations imposed by the county Highway Department require that no two of the billboards be within less than or equal to 5 miles of each other. You'd like to place billboards at a subset of the sites so as to maximize your total revenue, subject to this restriction.

Example: Suppose $M = 20, n = 4, \{x_1, x_2, x_3, x_4\} = \{6, 7, 12, 14\}$, and $\{r_1, r_2, r_3, r_4\} = \{5, 6, 5, 1\}$. Then the optimal solution would place billboards at x_1 and x_3 for a total revenue of 10.

Give an algorithm that takes an instance of this problem as input and returns the maximum total revenue that can be obtained from any valid subset of sites. The running time should be polynomial in terms of n , and as small as possible.

2. Maximizing Fun

A company is planning a party for its employees. The employees in the company are organized into a strict hierarchy, that is, a tree with the company president at the root. The organizers of the party have assigned a real number to each employee measuring how fun the employee is. In order to keep things social, there is one restriction on the guest list: an employee cannot attend the party if their immediate supervisor is present. On the other hand, the president of the company must attend the party, even though she has a negative fun rating; it's her company, after all. Give an algorithm that makes a guest list for the party that maximizes the sum of the fun ratings of the guests.

3. Moving on a Checkerboard

Suppose that you are given an $n \times n$ checkerboard and a checker. You must move the checker from the bottom edge of the board to the top edge of the board according to the following rule. At each step you may move the checker to one of three squares:

- 1) the square immediately above
- 2) the square that is one up and one to the left (but only if the checker is not already in the leftmost column)
- 3) the square that is one up and one to the right (but only if the checker is not already in the rightmost column)

Each time you move from square x to square y , you receive $p(x, y)$ dollars. You are given a list of the values $p(x, y)$ for each pair (x, y) for which a move from x to y is legal. Do not assume that $p(x, y)$ is positive.

Give an algorithm that figures out the set of moves that will move the checker from somewhere along the bottom edge to somewhere along the top edge while gathering as many dollars as possible. Your algorithm is free to pick any square along the bottom edge as a starting point and any square along the top edge as a destination in order to maximize the number of dollars gathered along the way. What is the running time of your algorithm?