Dynamic Programming - III

Tutorial Exercise#9

- Suppose the job of a firm is to manage the construction of billboards on the Surat-Dumas Gauravpath that runs east-west for $\mathbf{M}$ kms. The possible sites for the billboards are given by numbers $x_1, x_2, x_3, x_4, x_5, \ldots. x_n$ each in the interval $[0\ldots M]$. $x_i's$ are indicating the position of the billboards along the Gauravpath in kms, measured from its western end. If it is decided to place a billboard at location x_i, then the firm receives a revenue of $r_i>0$.

  Regulations of the SMC required that no two of the billboards be within less than or equal to t kms of each other. Thus, as part of the optimization problem, the firm has to decide where to place the billboards at a subset of the sites $x_1, x_2, x_3, x_4, x_5, \ldots. x_n$ so as to maximize the total revenue, subject to this constraint.

  Give an algorithm that takes in an instance of this problem as input and returns the maximum total revenue that can be obtained from any valid subset of sites. Give its running time also.

Tutorial Exercise#9...

- Illustration1:

  Suppose **M** = 20, n = 4, t=5

  Distances $[x_1, x_2, x_3, x_4]$ = [6,7,12,14]

  revenue $[r_1, r_2, r_3, r_4]$ = [5,6,5,1]

  Then, what would be the optimal solution ?

- Illustration2:

  Input : M = 20, separation distance t=5 kms

  Distances **x[] = {6, 7, 12, 13, 14}**

  revenue **[] = {5, 6, 5, 3, 1}** ....Then, Output: 10 ?

- Illustration3:

  Input : M = 15, separation distance t=2 kms

  Distances **x[] = {6, 9, 12, 14}**

  revenue **[] = {5, 6, 3, 7}** ....Then, Output : 18

Tutorial Exercise#10

- Suppose you own two stores, A and B. On each day you can be either at A or B. If you are currently at store A (or B) then moving to store B the next day (or A) will cost C amount of money. For each day i, i = 1, . . . , n, we are also given the profits $P^A(i)$ and $P^B(i)$ that you will make if you are store A or B on day i respectively. Give a schedule which tells where you should be on each day so that the overall money earned (profit minus the cost of moving between the stores) is maximized.
  - Approach:
    - Define two arrays TA[] and TB[]. TA[i] gives the most profitable schedule for days i, . . . , n given that we start at store A on day i. Define T B [i] similarly.
    - Write the recurrences… … …
- TA[i] = PA (i) + max(TA [i + 1], TB[i + 1] − C).

Tutorial Exercise#11

- You are going on a long trip. You start on the road at mile post 0. Along the way there are n hotels, at mile posts $a_1 < a_2 < \cdot \cdot \cdot < a_n$, where each a_i is measured from the starting point. The only places you are allowed to stop are at these hotels, but you can choose which of the hotels you stop at. You must stop at the final hotel (at distance an), which is your destination.
- You'd ideally like to travel 200 miles a day, but this may not be possible (depending on the spacing of the hotels). If you travel $\mathbf{x}$ miles during a day, the *penalty* for that day is $\mathbf{(200 - x)^2}$. You want to plan your trip so as to minimize the total penalty—that is, the sum, over all travel days, of the daily penalties.
- Give an efficient algorithm that determines the optimal sequence of hotels at which to stop.
- Solution:
- To get OPT(i), we consider all possible hotels j we can stay at the night before reaching hotel i. For each of these possibilities, the minimum penalty to reach i is the sum of:
  - the minimum penalty OPT(j) to reach j,
  - and the cost $(200 - (a_j - a_i))^2$ of a one-day trip from j to i.

Tutorial Exercise#13

- Design an efficient algorithm for finding the length of the longest path in a dag. (This problem is important both as a prototype of many other dynamic programming applications and in its own right because it determines the minimal time needed for completing a project comprising precedence-constrained tasks.) Show how to reduce the coin-row problem discussed in this section to the problem of finding a longest path in a dag.

Tutorial Exercise#14

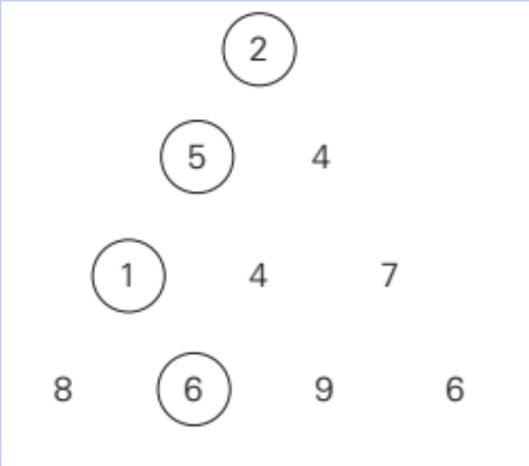
- Design a dynamic programming algorithm for the following problem. Find the maximum total sale price that can be obtained by cutting a rod of n units long into integer-length pieces if the sale price of a piece i units long is pi for for i = 1, 2, . . . , n. What are the time and space efficiencies of your algorithm?

Tutorial Exercise#15

- A chess rook can move horizontally or vertically to any square in the same row or in the same column of a chessboard. Find the number of shortest paths by which a rook can move from one corner of a chessboard to the diagonally opposite corner. The length of a path is measured by the number of squares it passes through, including the first and the last squares. Solve the problem
  - a. by a dynamic programming algorithm.
  - b. by using elementary combinatorics.

Tutorial Exercise#16

- Minimum-sum descent Problem: Some positive integers are arranged in an equilateral triangle with n numbers in its base like the one shown in the figure below for n = 4. The problem is to find the smallest sum in a descent from the triangle apex to its base through a sequence of adjacent numbers (shown in the figure by the circles). Design a dynamic programming algorithm for this problem and indicate its time efficiency.



Tutorial Exercise#17

- Virat Kohli on retirement, is considering opening a series of restaurants Kohli's along the Vadodara Mumbai Expressway (VME). The **n** possible locations are along a straight line, and the distances of these locations from the start of VME are, in kms and in increasing order, $m_1, m_2, \ldots, m_n$. The constraints are as follows:
  - At each location, Kohli's may open at most one restaurant. The expected profit from opening a restaurant at location i is p_i, where $p_i > 0$ and $i = 1,2,\ldots,n$.
  - Any two restaurants should be at least k kms apart, where k is a positive integer.

Design an efficient algorithm to compute the maximum expected total profit subject to the given constraints.