Assignment 7 – Dynamic Programming

Answer the questions in the boxes provided on the question sheets. If you run out of room for an answer, add a page to the end of the document.

Name: Ramapriya Wisc id: rranganath

Dynamic Programming

Do **NOT** write pseudocode when describing your dynamic programs. Rather give the Bellman Equation, describe the matrix, its axis and how to derive the desired solution from it.

1. *Kleinberg, Jon. Algorithm Design (p.313 q.2).*

   Suppose you are managing a consulting team and each week you have to choose one of two jobs for your team to undertake. The two jobs available to you each week are a low-stress job and a high-stress job.

   For week i, if you choose the low-stress job, you get paid ℓ_i dollars and, if you choose the high-stress job, you get paid h_i dollars. The difference with a high-stress job is that you can only schedule a high-stress job in week i if you have no job scheduled in week $i-1$.

   Given a sequence of n weeks, determine the schedule of maximum profit. The input is two sequences: $L := \langle \ell_1, \ell_2, \ldots, \ell_n \rangle$ and $H := \langle h_1, h_2, \ldots, h_n \rangle$ containing the (positive) value of the low and high jobs for each week. For Week 1, assume that you are able to schedule a high-stress job.

   (a) Show that the following algorithm does not correctly solve this problem.

Algorithm: JobSequence

Input : The low (L) and high (H) stress jobs.
Output: The jobs to schedule for the n weeks
for *Each week i* **do**
  if $h_{i+1} > \ell_i + \ell_{i+1}$ **then**
    Output "Week i: no job"
    Output "Week i+1: high-stress job"
    Continue with week i+2
  else
    Output "Week i: low-stress job"
    Continue with week i+1
  end
end

Solution: $L = \langle 1, 2, 3 \rangle$

$H = \langle 100, 5, 4 \rangle$

By algorithm, we have : $0 + h_2 + l_3 = 8$

But with schedule: $h_1 + l_2 + l_3 = 105$

(b) Give an efficient algorithm that takes in the sequences L and H and outputs the greatest possible profit. Prove that your algorithm is correct.

Solution: We have 1-d matrix. $m(j)$

Consider $m(j)$ is the maximum profit after week j.

Bellman equation:

$m(j) = \max\{m(j-1) + l_j,\ m(j-2) + h_j\}$

$m(0) = 0 \qquad m(1) = h_1$

prove correctness. Using induction: $j=2$, $m(2) = \max\{h_1 + l_2, h_2\}$ is optimal

for general j, by strong induction. we have $m(j-1)$, $m(j-2)$ is optimal

then $m(j)$ is optimal.

Complexity:

number of cells: $O(n)$, cost per cell: $O(1)$

Alg:

$m(0) = 0$, $m(1) = 1$

for $j = 2 \cdots n$

$m(j) = \max\{m(j-1) + l_j,\ m(j-2) + h_j\}$

end

2. *Kleinberg, Jon. Algorithm Design (p.315 q.4).*

Suppose you're running a small consulting company. You have clients in New York and clients in San Francisco. Each month you can be physically located in either New York or San Francisco, and the overall operating costs depend on the demands of your clients in a given month.

Given a sequence of n months, determine the work schedule that minimizes the operating costs, knowing that moving between locations from month i to month $i-1$ incurs a fixed moving cost of M. The input consists of two sequences N and S consisting of the operating costs when based in New York and San Francisco, respectively. For month 1, you can start in either city without a moving cost.

(a) Give an example of an instance where it is optimal to move at least 3 times. Explain where and why the optimal must move.

Solution:

$N = \langle 0, 10M, 0, 10M, 0, 10M \rangle$

$S = \langle 10M, 0, 10M, 0, 10M, 10 \rangle$

As shown in example, it's optimal to switch to 0 operating cost each month.

(b) Show that the following algorithm does not correctly solve this problem.

Algorithm: WorkLocSeq

Input : The NY (N) and SF (S) operating costs.
Output: The locations to work the n months

```
for Each month i do
    if N_i < S_i then
        Output "Month i: NY"
    else
        Output "Month i: SF"
    end
end
```

Solution:

$N: \langle \frac{M}{2}, M, \frac{M}{2} \rangle$

$S: \langle M, \frac{M}{2}, M \rangle$

the algorithm would cost: $\frac{M}{2} + M + \frac{M}{2} + M + \frac{M}{2} = \frac{7}{2}M$

But if we stay in NY for 3 months, we cost

$\frac{M}{2} + M + \frac{M}{2} = 2M$

(c) Give an efficient algorithm that takes in the sequences N and S and outputs the value of the optimal solution. Prove that your algorithm is correct.

Solution:

We have 2-d matrix. $m(L, j)$. $m(j)$ is minimum cost

Consider $m(L, j)$ is the minimum cost after month j.

when worked at location $L \in \{NY, SF\}$ at month j

Bellman equation:

$$m(NY, j) = \min\{ m(NY, j-1) + N_j,\ m(SF, j-1) + N_j + M \}$$

$$m(SF, j) = \min\{ m(SF, j-1) + S_j,\ m(NY, j-1) + S_j + M \}$$

$$m(j) = \min\{ m(NY, j),\ m(SF, j) \}$$

where $m(NY, 1) = N_1$ $m(SF, 1) = S_1$

prove correctness. Using induction: $j=2$.

$$m(NY, 2) = \min\{ m(NY, 1) + N_2,\ m(SF, 1) + N_2 + M \}$$

$$m(SF, 2) = \min\{ m(SF, 2) + S_2,\ m(NY, 2) + S_2 + M \}$$

$m(2)$ is min

for general j. By induction.

we have $m(NY, j)$ is optimal when we worked at NY at month j

$m(SF, j)$ is optimal similarly.

then $m(j)$ is optimal.

3. *Kleinberg, Jon. Algorithm Design (p.333, q.26).*

Consider the following inventory problem. You are running a company that sells trucks and predictions tell you the quantity of sales to expect over the next n months. Let d_i denote the number of sales you expect in month i. We'll assume that all sales happen at the beginning of the month, and trucks that are not sold are stored until the beginning of the next month. You can store at most s trucks, and it costs c to store a single truck for a month. You receive shipments of trucks by placing orders for them, and there is a fixed ordering fee k each time you place an order (regardless of the number of trucks you order). You start out with no trucks. The problem is to design an algorithm that decides how to place orders so that you satisfy all the demands $\{d_i\}$, and minimize the costs. In summary:

- There are two parts to the cost: (1) storage cost of c for every truck on hand; and (2) ordering fees of k for every order placed.
- In each month, you need enough trucks to satisfy the demand d_i, but the number left over after satisfying the demand for the month should not exceed the inventory limit s.

Give an algorithm that solves this problem in time that is polynomial in n and s. Prove that the algorithm is optimal.

Solution:

$m(j,s)$ is the cost after j month. where there are S trucks in the inventory at month $j-1$

$$m(n,s) = \min\{\max\{c(s-d_n),0\},k\}$$

$$m(n,s-1) = \min\{\max\{c(s-1-d_n),0\},k\}$$

$\vdots$

$m(n,0)$

Bellman equation:

$$m(j,s) = m(j+1,s) - k - s\cdot c$$

$$m(j,s-1) = m(j+1,s-1) - k - (s-1)c$$

$\vdots$

$m(j,0)$

$$m(j,s) = m(j+1,s-a) + k\cdot\mathbb{1}(a<d_j) + c\cdot(s-a)$$

$$\begin{array}{c|c} & 0 \ \cdots \ s \\ \begin{matrix}1\\ \vdots\\ n\end{matrix} & \left[\begin{matrix}\ \\ \ \\ \ \end{matrix}\qquad\qquad\right] \end{array}$$