Book 6.4: Business Costs (East vs West)

Background

Suppose you're running a lightweight consulting business – just you, two associates, and some rented equipment. Your clients are distributed between the East COast and the West Coast, and this leads to the following question.

Each month, you can either run your business from an office in New York (NY) or from an office in San Francisco (SF). In month i, you'll incure anoperating cost of N_i if you run the business out of NY; you'll incur an operating cost of S_i if you run the business out of SF. (It depends on the distribution of client demands for that month.)

However, if you run the business out of one city in month i, and then out of the other city in month $i+1$, then you incure a fixed moving cost of M to switch base offices.

Given a sequence of n months, a plan is a sequence of n locations – each one equal to either NY or SF – such that the i^{th} location indicates the city in which you will be based in the i^{th} month. THe cost of a plan is the sum of the operating costs for each of the n months. plus a moving cost of M for each time you switch cities. The plan can begin in either city.

1 Part A

Show that the following algorithm does not correctly solve this problem by giving an instance on which it does not return the correct answer.

Algorithm 1 Given From Part A

```
procedure MAIN
    for i = 1 to n do
        if $N_i$<$S_i$ then
            Output "NY in Month i"
        else
            Output "SF in Month i"
```

Solution

Take the example: with $n = 4$ and $M = 10$

Loaction	Month 1	Month 2	Month 3	Month 4
New York	1	3	1	3
San Francisco	3	1	4	1

Following Algorithm A we would get NY, SF, NY, SF = 1 + (10 + 1) + (10 + 1) + (10 + 1) = 34, however the correct answer is NY, NY, NY, NY = 1 + 3 + 1 + 3 = 8 and 5 <34 $\implies$ Algorithm A does not give us the correct answer.

2 Part B

Give an example of an instance in which every optimal plan must move (i.e., change locations) at least three times. Provide a brief explanation, saying why your example has this property.

Solution

Let us use the example from above except instead of $M = 10$, let $M = 1$ $\therefore$ the cheapest route would be NY, SF, NY, SF = 1 + (1 + 1) + (1 + 1) + (1 + 1) = 7. This is the least expensive since it costs less to work in these cities with the added travel cost at each stage than any other possible path.

3 Part C

Give an efficient algorithm that takes values for n, M, and sequences of operating costs $N_1, ..., N_n$ and $S_1, ..., S_n$, and returns the cost of an optimal plan.

Solution

3.1 Mathematical Formulation

Given a value for the moving cost M, and sequences of operating costs over n months as $N_1, ..., N_n$ and $S_1, ..., S_n$, find the minimal cost for a plan of length n.

3.2 Algorithm

Important Confusing Data Structures:

- int[n] **optNY** : Keeps track of the minimum cost of plan terminating in New York on month i
- int[n] **optSF** : Keeps track of the minimum cost of plan terminating in San Francisco on month i

We will implement the below rules in our algorithm to produce the solution wehere cNY and cSF are the respective costs of operating in New York and Sanfrancisco in month i.

$$optNY(i) = min\begin{cases} optNY(i-1) + cNY(i) \\ optSF(i-1) + cNY(i) + M \end{cases}$$

$$optSF(i) = min\begin{cases} optNY(i-1) + cSF(i) + M \\ optSF(i-1) + cSF(i) \end{cases}$$

What this means is that the minimal total cost to operate in NY at month i will always be the cost of operating in New York at month i plus the either the total cost of operating in either NY or SF in month $i-1$. The way we determine which to pick is to ask wheather the previous total cost in NY is less than the previous total cost in SF with the additional travel fee added in. Likewise, the same calculation can be made for the minimal total cost to operate in SF at month i.

Once we have figured these out, we know that any path we take will end in either NY or SF at month n so we just have to take the minimal of these to get our result. i.e. choose min(optNY(n), optSF(n))

Algorithm 2 True Solution

```
procedure MAIN(cNY, cSf, M , n)
    $optNY, optSF$ ← initialized where $optNY(0)$ ← cNY(0); $optSF(0)$ ← cSF(0)
    for $i \in [1,(n-1)]$ do
        $vNY$ ← cNY(i) + optNY(i-1) // value coming from NY
        $vSF$ ← cNY(i) + optSF(i-1) // value coming from SF
        if vNY ≤ vSF + M then
            $optNY(i)$ ← vNY
        else
            $optNY(i)$ ← vSF + M
        vNY, vSF += SF(i) - NY(i) // switch to chosing for SF
        if vSF ≤ vNY + M then
            $optSF(i)$ ← vSF
        else
            $optSF(i)$ ← vNY + M
    PRINT(min(optNY(n-1), optSF(n-1)))
```

3.3 Correctness

For the following proofs, please note that the information n : number of test cases, M : cost for relocation, $cNY(x), 1 \leq x \leq n$: the cost to run a business in New York in month x, and $cSF(y), 1 \leq y \leq n$: the cost to run a business in San Francisco in month y.

Proposition 1.

The relationships below will give us the lowest total costs to work in New York or San Francisco in month i. Note: optNY(1) = cNY(1) and optSF(1) = cSF(1)

$$optNY(i) = min \begin{cases} optNY(i-1) + cNY(i) \\ optSF(i-1) + cNY(i) + M \end{cases}$$

$$optSF(i) = min \begin{cases} optNY(i-1) + cSF(i) + M \\ optSF(i-1) + cSF(i) \end{cases}$$

Proof.

One thing to keep in mind here is determining wheather coming from New York or San Francisco is not dependent on the cost of what it costs to work in the city during the current month i. i.e. (W.L.O.G) optNY(i-1) + cNY(i) <optSF(i-1) + cNY(i) + M $\implies$ optNY(i-1) < optSF(i-1) + M

For this proof we will use induction. We start with <u>i = 2</u>: at this point we have only two options and that is to originate from optNY(2-1 = 1) = cNY(1) or optSF(2-1 = 1) = cSF(1); for optNY(2) then we have to consider staying in the same city (cNY(1)) or relocate (cSF(1) + M), so naturally we pick the smaller $\implies$ optNY(2) = min(cNY(1), [cSF(1) + M]); similarly we will have optSF(2) = min(cSF(1), [cNY(1) + M]). We see that both of these hold.

Now lets assume that $i = k$ holds so we have optNY(k) and optSF(k) for some $k, 1<k<n$. Therefore we now will extend and check i = k + 1 we see that our only options for optNY(k+1) is originating from either New York or San Francisco in month k. Since optNY(k) represents the cheapest way to end working in New York in month k [mirrored for optSF(k)], we simply have to compare the prices and consider staying in the same city (cNY(k)) or relocate (cSF(k) + M), so naturally we pick the smaller $\implies$ optNY(k+1) = min(cNY(k), [cSF(k) + M]); similarly we will have optSF(k+1) = min(cSF(k), [cNY(k) + M]). We see that letting k = i-1 this is the exact relationship. □

Corollary 2.
The cost of the cheapest (optimal) plan is min(optNY(n), optSF(n)).

Proof.
Since proposition 1 holds we see that we can build the tables optNY() and optSF(). Therefore since the cheapest option must terminate in the n^{th} and there are only two options of working in either New York or San Francisco during this month, and by the above we have the cheapest total costs to a plan which ends working in New York and San Francisco in month n. If we take the least one then we will have the minimal plan. □

3.4 Analysis

Proposition 3. *The space complexity of this algorithm is **O(N)***

Proof.
This is due to the fact that all of our data is stored in data structures:

- cNY: cost to run business in New York in month $i \implies size(N)$
- cSF: cost to run business in San Francisco in month $i \implies size(N)$
- optNY: storing minimal cost to run in New York in month i taking the past into consideration $\implies size(N)$
- optSF: storing minimal cost to run in San Francisco in month i taking the past into consideration $\implies size(N)$

Summing yeilds $4 \cdot N$

Giving us a space complexity of **O(N)**

□

Proposition 4. *The time complexity of this algorithm is* ***O(N)***

Proof. This is the case because our algorithm simply will linearly look through the given data of size N once, at each stage looking at the previous entry O(1) four times.

Giving us a time complexity of **O(N)**

□

3.5 An Example

Let $n = 4$ and $M = 1$ and

	Month 1	Month 2	Month 3	Month 4
cNY	1	3	1	3
cSF	3	1	4	1

We will initialize the tables optNY and optSF then as such

	Month 1	Month 2	Month 3	Month 4
optNY	1	-	-	-
optSF	3	-	-	-

Now we will emplore our relationships and we see the table fills out as follows for the i = 2

	Month 1	Month 2	Month 3	Month 4
optNY	1	min(1 + 3, 3 + 3 + 1) = 4	-	-
optSF	3	min(3 + 1, 1 + 1 + 1) = 3	-	-

Then for i = 3

	Month 1	Month 2	Month 3	Month 4
optNY	1	4	min(4 + 1, 3 + 1 + 1) = 5	-
optSF	3	3	min(3 + 4, 4 + 4 + 1) = 7	-

And for i = 4 = n

	Month 1	Month 2	Month 3	Month 4
optNY	1	4	5	min(5 + 3, 7 + 3 + 1) = 8
optSF	3	3	7	min(7 + 1, 5 + 1 + 1) = 7

So now that we have filled out the tables completely, to determine the optimal cost for our plan we take the min(optNY(n),optSF(n)) $\implies$ min(8,7) = 7