Middle East Technical University Department of Computer Engineering

CENG 567 - Second Program

Design and Analysis of Algorithms

Spring 2019-2020

Take Home Exam 3 (THE-3)

Due date: 16 May 2020 - 23:55

1 Dynamic Programming Approach for ATM Cash Replenishment Optimization Problem

In Automated Telling Machine (ATM) cash replenishment problem, banks aim to reduce the number of out-of-cash ATMs and duration of out-of-cash status. On the other hand, they want to reduce the cost of cash replenishment, as well. We assume that reliable forecasts are already obtained for the amount of cash needed in ATMs.

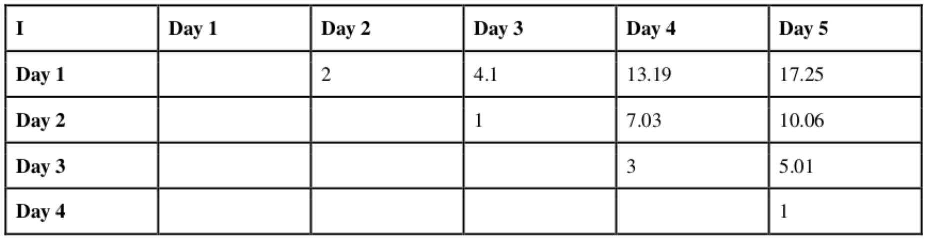
The inputs of the cash replenishment optimization problem (ROP) are as follows:

- predicted withdrawal amounts for N consecutive days for an ATM,
- the daily interest rate,
- cash transportation/loading cost.

However, we can pre-calculate the total interest cost for the amount corresponding to each sub-period between the first day and the day N using the first 2 items.

Accumulated Interest Cost (I[i, j]): Total interest cost incurred for the amount corresponding to the predicted withdrawals of an ATM for a replenishment period [i, j] (i.e, from day i to day j) for a given daily interest rate.

1.1 Example 1



I	**Day 1**	**Day 2**	**Day 3**	**Day 4**	**Day 5**
Day 1		2	4.1	13.19	17.25
Day 2			1	7.03	10.06
Day 3				3	5.01
Day 4					1

Figure 1: Accumulated Interest Costs

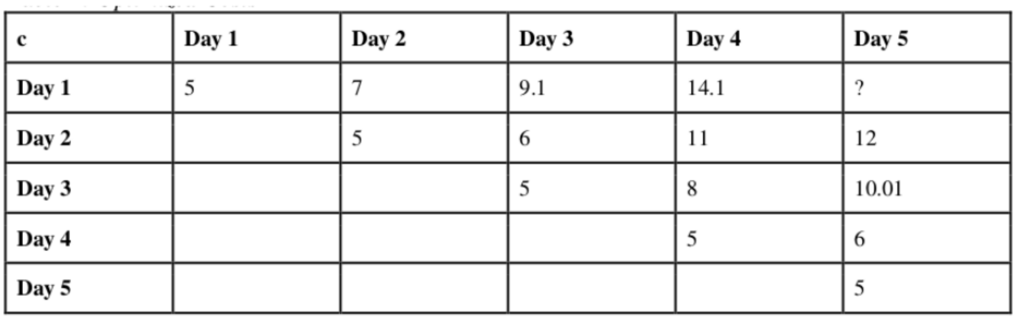
Simplified definition of the ROP problem, which uses the accumulated interest cost (I [i, j] matrix for every pair of days from i to j) and a fixed loading cost α, the recurrence relation of ROP can be defined as follows:

$$c[i,j] = min \begin{cases} min_{i \le k < j}(c[i,k] + c[k+1,j]) \\ \sum_{r=i}^{j} I[i,j] + \alpha \end{cases}$$

Figure 2: Recurrence relation of ROP

In this definition, c[i, j] is the minimized replenishment cost for an ATM from day i to j (including j). As it is done for the MCP problem, the actual replenishment days producing this minimum cost can easily be obtained by storing the k values during the calculation of the cost matrix.

Example solution of above input I and α=5:



c	Day 1	Day 2	Day 3	Day 4	Day 5
Day 1	5	7	9.1	14.1	?
Day 2		5	6	11	12
Day 3			5	8	10.01
Day 4				5	6
Day 5					5

Figure 3: Optimized Costs

The final entry to be calculated is c[1,5], which is the result of this problem instance. Calculations for c[1,5] requires choosing the minimum of the results of Equations 8-12.

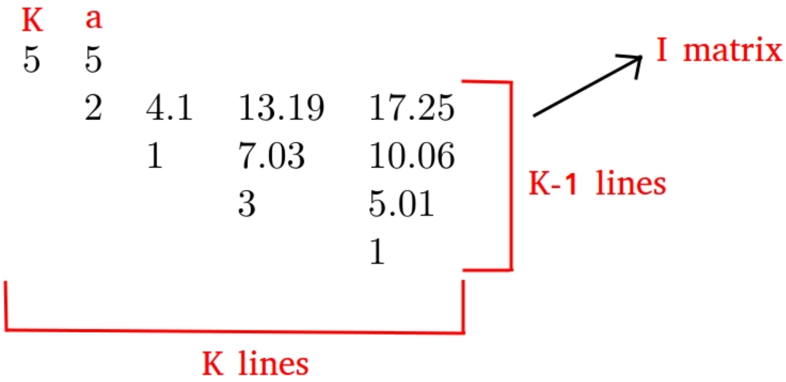
```
(12345) = α + I[1,5] = 5 + 17.25 = 22.25 (8)
(1)(2345) = c[1,1] + c[2,5] = 5 + 12 = 17 (9)
(12)(345) = c[1,2] + c[3,5] = 7 + 10.01 = 17.01 (10)
(123)(45) = c[1,3] + c[4,5] = 9.1 + 6 = 15.1 (11)
(1234)(5) = c[1,4] + c[5,5] = 14.1 + 5 = 19.1 (12)
```

Figure 4: Equations 8-12

Among these 5 choices, the solution that corresponds to (123)(45) is the optimum one for c[1,5] whose value is 15.1. This solution means that we should load the first three days' amount on day 1, and the last two days' amount on day 4.

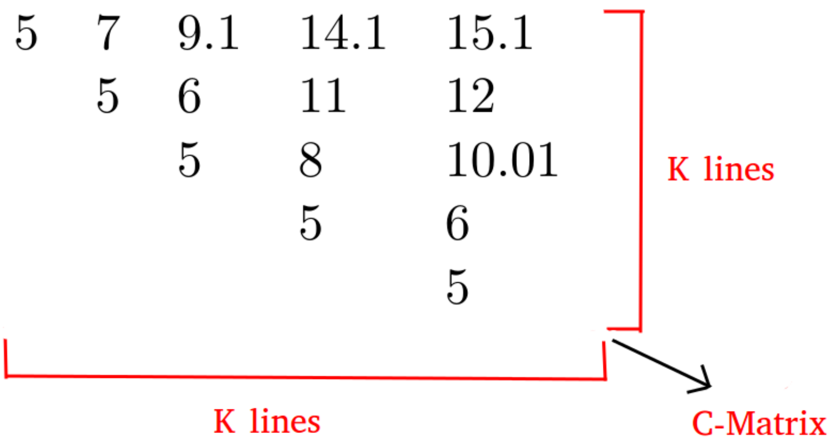
2 Input/Output Specifications

Your input file will consist K α in the first line and and I matrix ($K-1$ x K) in the following lines. Your output should consist C matrix (K x K). For the empty places you should output e character (examine the given example inputs and outputs. The empty lines in I should be calculated as 0 (for example in the following example $I[1,1]$ is 0.
K will be an integer between 3 and 1000.
α and the matrix values will be between 1 and 30 (they are real numbers not guaranteed to be integer)



```
5   5
    2   4.1   13.19   17.25
        1     7.03    10.06
              3       5.01
                      1
```

Figure 5: Shows an example input file (inp1). The first line consist of K and alpha and the following lines consist of the I matrix. Notice that there will be 'e' characters instead of empty characters in the input file.



```
5   7   9.1   14.1   15.1
    5   6     11     12
        5     8      10.01
              5      6
                     5
```

Figure 6: Shows an example output file (out1). It consist of the C matrix. Notice that there will be 'e' characters instead of empty characters in the output file.

3 Deliverables

- You can use C++,Java or Python. You will only upload one file named as the1.cpp (you can change the part .cpp to any extension you want according to language you use). Do not submit any other files.
- Submit the file through the ODTUClass system before the given deadline. **16 May 2020 - 23:55**

4 Regulations

- **Cheating:** This homework should be done individually. Copying another one's solution as it is, copying another one's solution and modifying it, or giving your own solution to someone else will be respected as cheating. Taking code directly from internet is also regarded as cheating. Grader of this homework has the right to request defense from those he suspects to be cheaters. Those whose defenses are not accepted will be subject to disciplinary actions according to the university regulations. Please note that defenses will only be taken in written way.
- **Odtuclass Discussions:** You should follow the odtuclass for discussions and possible updates on a daily basis.
- **Late Submission:** Not allowed.