Week 11: Assignment 1

The due date for submitting this assignment has passed.

Due on 2022-10-19, 23:59 IST.

Assignment submitted on 2022-10-19, 23:04 IST

1) You have built a popcorn dispenser for people who love precision. The dispenser has has a display which initially shows the number 0. It further has two buttons:

(1) INC
(2) DEC

Pressing INC increases the number on the display by 1.

Pressing DEC decreases the number on the display by 1 (if the number on the display is 0, pressing DEC has no effect).

At any point, you can bring a bowl under the dispenser. This will trigger a circuit and precisely X popcorns will be dispensed into the bowl, where X is the number currently on display. Note that the display does **not change** when the popcorns are being dispensed.

Today you have a queue of N groups of P people. Each person has a desired number of popcorns that they want to take with them.

You can serve the people in a group in any order you want, but you cannot change the order of the groups.

Specifically, you must serve all customers from the i-th group before addressing any from the (i+1)-th group. In between handling two customers, if those two customers have different amounts of popcorn that they want, you need to use the buttons on the machine.

The number on the dispenser can be left at any number after all customers have been served, since it will reset automatically at midnight.

The question we want to solve is the following: If you order the customers of each group optimally, what is the minimum number of button presses you need?

Here's a worked out example. Suppose N = 1 and P = 2.

Let's say the first customer wants 20 popcorns and the second customer wants 50. You need to press the button 50 times total (first 20, then 30) if you have the first customer serve first and then the second. In the opposite order, you would need to press the button 80 times total (first 50, then 30). Clearly, the first order is the better one.

Now suppose N = 1, P = 3, and the individual demands are 20, 20, 30. What is the optimal number of button presses?

30

Yes, the answer is correct.
Score: 1

Accepted Answers:
(Type: Numeric) 30

1 point

2) Now suppose N = 3 and P = 3 and the demands of the three groups are as given below:

Group 1: 30 10 40
Group 2: 20 50 60
Group 3: 60 60 50

110

Yes, the answer is correct.
Score: 1

Targeted Feedback:

An an optimal way to use the pump is:

- *press up 10 times, setting the pump to 10; pump the product (from customer 1) that needs 10 pascals,*
- *press up 30 times, setting the pump to 40; pump the product (from customer 1) that needs 40 pascals,*
- *press down 10 times, setting the pump to 30; pump the product (from customer 1) that needs 30 pascals,*
- *press down 10 times, setting the pump to 20; pump the product (from customer 1) that needs 20 pascals,*

- *press up 30 times, setting the pump to 50; pump the product (from customer 2) that needs 50 pascals,*
- *press up 10 times, setting the pump to 60; pump the product (from customer 2) and the two products (from customer 3) that need 60 pascals, and finally*
- *press down 10 times, setting the pump to 50; pump the product (from customer 3) that needs 50 pascals.*

This is a total of 110 button presses.

Accepted Answers:
(Type: Numeric) 110

1 point

3) If within a group the largest demand for popcorn is q and the smallest demand is p, what is the minimum number of times you HAVE to press the button to serve everyone in the group? ***2 points***

Do not assume that the display is at 0 when you start working with this group.

- q
- p
- q-p
- p+q

Yes, the answer is correct.
Score: 2
Accepted Answers:
q-p

4) Consider the following approach. ***2 points***

You decide to serve each group either in increasing or decreasing order of demand. You try all possibilities of which groups will be served in increasing order of demand and which groups will be served in decreasing order of demand.

What is the complexity of this approach?

To make sure we understand the approach, consider the following example, with 2 groups of three people:

Group 1: 30 10 40
Group 2: 20 50 60

There are four possibilities now:

A. Group 1 is served in increasing order, Group 2 in decreasing order.
B. Group 1 is served in decreasing order, Group 2 in increasing order.

C. Both groups are served in increasing order.
D. Both groups are served in decreasing order.

For method A, the number of button presses will be: 100
For method B, the number of button presses will be: 120
For method C, the number of button presses will be: 100
For method D, the number of button presses will be: 160

- (•) O(2^N)
- ( ) O(N!)
- ( ) O(N^2)
- ( ) O(N log N)

Yes, the answer is correct.
Score: 2
Accepted Answers:
O(2^N)

5) Consider the approach from the previous question: trying all possibilities of which groups will be served in increasing order of demand and which groups will be served in decreasing order of demand. Is it possible that the answer you find is not optimal? ***2 points***

- ( ) Yes, this approach may miss out on an optimal method.
- (•) No, there is always an optimal solution where every group of customers are served either in increasing or decreasing order of demand.

Yes, the answer is correct.
Score: 2
Accepted Answers:
No, there is always an optimal solution where every group of customers are served either in increasing or decreasing order of demand.

6) Suppose we continue to take the approach of serving every group in increasing or decreasing order of demand. Based on this, consider the following improvisation on the brute force strategy. ***2 points***

Let DP[i,0] be the answer after processing i groups of customers where the customers for the last group are arranged in increasing order.

Let DP[i,1] be the answer after processing i groups of customers where the customers for the last group are arranged in decreasing order.

Let P[i] denote the largest demand in the i-th group.

Let Q[i] denote the smallest demand in the i-th group.

Let F[i] denote P[i] - Q[i].

Now consider the following recurrence.

DP[i+1,0] = min(DP[i,0] + |P[i]-Q[i+1]| + F[i+1], DP[i,1] + |Q[i]-Q[i+1]| + F[i+1])

What should the recurrence for DP[i+1,1] be?

- min(DP[i,0] + |Q[i]-Q[i+1]| + F[i+1], DP[i,1] + |P[i]-Q[i+1]| + F[i+1])
- min(DP[i,0] + |P[i]-P[i+1]| + F[i+1], DP[i,1] + |Q[i]-P[i+1]| + F[i+1])
- min(DP[i,0] + |P[i]-P[i+1]| + F[i+1], DP[i,1] + |Q[i]-Q[i+1]| + F[i+1])
- min(DP[i,0] + |Q[i]-Q[i+1]| + F[i+1], DP[i,1] + |P[i]-P[i+1]| + F[i+1])

Yes, the answer is correct.
Score: 2

Accepted Answers:
min(DP[i,0] + |P[i]-P[i+1]| + F[i+1], DP[i,1] + |Q[i]-P[i+1]| + F[i+1])