Code Jam 2022 - Round 1B

Analysis: Controlled Inflation

The key observation is that for each customer, either the increasing or decreasing order of target pressures will produce an optimal solution. That is, no other permutation of products will result in a strictly lower number of button presses. Let's prove this.

First, note that this is true for each individual customer i. At one point in the process, the pump will be at the minimal pressure $\mathrm{Min}_i = \min_{j=1..\mathbf{P}} \mathbf{X_{i,j}}$. At another point, the pump will be at the maximal pressure $\mathrm{Max}_i = \max_{j=1..\mathbf{P}} \mathbf{X_{i,j}}$. This means that we have to reach both of them, so we will always have to press the buttons at least $\mathrm{Max}_i - \mathrm{Min}_i$ times, no matter the order. This can be achieved by arranging the products in either increasing or decreasing order of their target pressures.

Now, the pressure also has to be adjusted between customers, which requires pressing the buttons. Let's see why all other product orders do not improve the answer by consindering how many button presses between customers we can save. While processing the i-th customer, the pump will be at Min_i at one point, at Max_i at another point, and finally we will leave it at the pressure of the last product, $\mathbf{X_{i,last}}$. This requires at least $(\mathrm{Max}_i - \mathrm{Min}_i) + (\mathrm{Max}_i - \mathbf{X_{i,last}})$ button presses, but the potential saving is at most $(\mathrm{Max}_i - \mathbf{X_{i,last}})$, which is the least amount of additional button presses we needed to do. If the pump reaches the maximum pressure before reaching the minimum pressure, a similar relationship holds, meaning this different order does not improve the answer.

Now that we know that it is enough to consider only the increasing and decreasing orders, we will only keep track of Min_i and Max_i for each customer. For the test set with the visible verdict, it is enough to check all $2^{\mathbf{N}} \le 1024$ possibilities of choosing the increasing or decreasing order for each customer and simulate the process.

To do this more efficiently, we can use dynamic programming. Let $dp_{i,0}$ be the answer after processing i customers where the products for the last customers are arranged in increasing order. Similarly, let $dp_{i,1}$ be the answer after the first i customers with the products for the last one arranged in decreasing order. We will also keep track of the pressure we left the pump at, l_0 and l_1 for the increasing and decreasing orders corresponsindly. Clearly, $dp_{0,0} = dp_{0,1} = 0$ and $l_0 = l_1 = 0$ (the starting pressure). Now, assume dp is calculated up to i. The following equations give the values for the next customer:

$$dp_{i+1,0} = \min \begin{pmatrix} dp_{i,0} + |l_0 - \mathrm{Min}_{i+1}| + (\mathrm{Max}_{i+1} - \mathrm{Min}_{i+1}), \\ dp_{i,1} + |l_1 - \mathrm{Min}_{i+1}| + (\mathrm{Max}_{i+1} - \mathrm{Min}_{i+1}) \end{pmatrix}$$

$$dp_{i+1,1} = \min \begin{pmatrix} dp_{i,0} + |l_0 - \mathrm{Max}_{i+1}| + (\mathrm{Max}_{i+1} - \mathrm{Min}_{i+1}), \\ dp_{i,1} + |l_1 - \mathrm{Max}_{i+1}| + (\mathrm{Max}_{i+1} - \mathrm{Min}_{i+1}) \end{pmatrix}$$

And update the last pressures: $l_0 = \mathrm{Max}_{i+1}$, $l_1 = \mathrm{Min}_{i+1}$.