

## Problem E

### Parks of Jakarta

There are three parks in Jakarta, called Park 1, Park 2, and Park 3. The new governor of Jakarta wants to add decorations to these parks by adding bricks. These bricks are going to be stacked on top of another on these three parks. There are  $N$  bricks in Jakarta, indexed from 1 to  $N$ . A brick with index  $i$  is smaller than a brick with index  $i + 1$ . We do not want to put a bigger brick on top of a smaller brick. Therefore, we can put a brick with index  $i$  on top of a brick with index  $j$  if and only if  $i < j$ .

We define a brick configuration as a distribution of these  $N$  bricks on the three parks. Note that there must only be one stack of bricks in each park, and the bricks must be stacked in the order as explained on the previous paragraph. For example, if  $N = 3$ , then a possible brick configuration is: brick 1 (on the top) and brick 2 (on the bottom) are in park 1, no brick is in park 2, and brick 3 is in park 3.

We can change a brick configuration to another by doing an operation, which we will:

- Pick two different parks  $i$  and  $j$ . Take the topmost brick on the stack in park  $i$  and put it on the top of the stack in park  $j$ . If there is no brick in park  $i$ , then this operation is not valid. If this operation causes the stack in park  $j$  to violate the stack order condition, then this operation is also not valid.
- Since we need to rent a truck to transport the brick from park  $i$  to park  $j$ , we need to pay  $R_{i,j}$  units of money for this operation. Note that the cost of transporting a brick from park  $i$  to park  $j$  might be different with the cost of transporting a brick from park  $j$  to park  $i$ .

Initially, the bricks are in some brick configuration, which we will call the initial brick configuration. The new governor of Jakarta wants to try  $M$  brick configurations. Therefore, from the initial brick configuration, we must do several operations such that each of the  $M$  brick configurations (in any order) wanted by the new governor is seen at least once. In the end, the new governor also wants us to put all the bricks in (any) one park. We want to minimise the total cost to satisfy his demand.

Formally, let  $G_1, G_2, \dots, G_M$  be the brick configurations wanted by the new governor. We want to construct a sequence of brick configurations  $C_0, C_1, \dots, C_k$  ( $k \geq 0$ ) with the minimum cost where:

- $C_0$  is the initial brick configuration
- There exists a sequence of integers  $x_1, x_2, \dots, x_M$  ( $0 \leq x_i \leq k$ ) where  $C_{x_i} = G_i$ . Note that the sequence is not necessarily increasing.
- $C_k$  is a brick configuration where all the bricks are stacked in (any) one of the parks.
- For all  $0 \leq i < k$ , we can achieve  $C_{i+1}$  from  $C_i$  by doing a single operation (described in paragraph three above).
- The cost of this sequence is the sum of cost needed to change  $C_i$  to  $C_{i+1}$  for all  $0 \leq i < k$ .

Let us help the new governor of Jakarta!

### Input

The first line contains two integers:  $N$   $M$  ( $1 \leq N \leq 40$ ;  $0 \leq M \leq 16$ ) in a line denoting the number of bricks and the number of configurations. The next three lines, each contains three integers. The  $j$ -th

integer on the  $i$ -th line is  $R_{i,j}$  ( $0 \leq R_{i,j} \leq 1000$ ;  $R_{i,i} = 0$ ) denoting the cost of moving a brick from the  $i$ -th city to the  $j$ -th city. The next three lines denote the initial brick configuration, with the format explained in the next paragraph. The next  $M$  blocks describe the configurations wanted by the new governor, where each block contains three lines, with the format of each configuration explained in the next paragraph.

Each configuration is denoted by three lines. The  $i$ -th line begins with an integer  $K$  ( $0 \leq K \leq N$ ) denoting the number of bricks in park  $i$ , followed by  $K$  integers:  $A_1 A_2 \dots A_K$  ( $1 \leq A_i \leq N$ ) denoting the brick indices in park  $i$ . It is guaranteed that  $A_i < A_j$  for all  $1 \leq i < j \leq K$ . It is also guaranteed that each of the integers between 1 and  $N$ , inclusive, appears exactly once on the union of the array  $A$  on the three lines. For example, the sample brick configuration given on the second paragraph will be illustrated in the first sample input.

## Output

The output contains the minimum total cost (in units of money) to satisfy the new governor's demand, in a line.

Sample Input	Output for Sample Input	Sample Input	Output for Sample Input
<pre> 3 0 0 1 1000 1 0 1 1000 1 0 2 1 2 0 1 3 </pre>	5	<pre> 3 2 0 2 2 2 0 2 2 2 0 2 1 2 1 3 0 3 1 2 3 0 0 0 0 3 1 2 3 </pre>	22

### Explanation for the 1<sup>st</sup> sample case

In the first sample, we initially have brick 1 and brick 2 in park 1, and brick 3 in park 3. Since  $M = 0$ , our goal is to only reach a brick configuration where all bricks are in the same (any) park. Therefore, we can do the following operations:

- Move brick 3 from park 3 to park 2. This costs 1 unit of money.
- Move brick 1 from park 1 to park 2. This costs 1 unit of money.
- Move brick 1 from park 2 to park 3. This costs 1 unit of money.
- Move brick 2 from park 1 to park 2. This costs 1 unit of money.
- Move brick 1 from park 3 to park 2. This costs 1 unit of money.

Therefore, all of the bricks are now in park 2 and we spend 5 units of money. There is no solution that costs less than 5 units of money. Note that we do not necessarily minimise the number of operations.

### Explanation for the 2<sup>nd</sup> sample case

In the second sample, the optimal solution can be achieved if we satisfy the second configuration first before the first one. From the initial configuration, we can get to the second configuration with 4 operations with the total cost of 8 units. From the second configuration, we can get to the first configuration with 7 operations with the total cost of 14 units. Since the first configuration already has all the bricks stacked in one park, we don't need any additional operation. Therefore, the total cost is 22 units.