OTHS Coding Competition 1 (Mock CCC) J5 - Scavenger Hunt

Time limit: 1.0s
Java: 3.0s
Python: 5.0s

Memory limit: 512M

You are participating in a scavenger hunt in a city with N buildings (1-indexed) and M **two-way** roads connecting buildings a_i and b_i, taking c_i minutes to travel. You are currently in building 1 and your goal is to obtain K items labelled from 1 to K which are scattered across the city, **in order**. The i^{th} item will be present in k_i buildings. Building 1 is guaranteed to never contain any items and a building may contain more than 1 item. For each item, you also have the option to stand still and create it yourself in m_i minutes. What is the minimum amount of time you need to obtain all K items?

Constraints

$1 \leq N, M \leq 20\,000$

$1 \leq a_i, b_i \leq N$

$a_i \neq b_i$

$1 \leq c_i \leq 10^6$

$1 \leq m_i \leq 10^9$

$1 \leq K \leq 30$

$1 \leq k_i < N$

Building 1 will never contain any items.

Subtask 1 [4/15]

$k_i = 1$

A building will contain at most 1 item.

Subtask 2 [4/15]

$m_i = 10^9$

$c_i \leq 10^3$

Subtask 3 [7/15]

No additional constraints.

Input Specification

The first line contains 3 integers, N, M, and K, the number of buildings, the number of roads, and the number of items you need to collect, respectively.

The next line contains K space separated integers, $m_1, \ldots, m_K$, where m_i is the time it takes to build item i yourself.

The next line contains K space separated integers, $k_1, \ldots, k_K$, where k_i is the number of buildings that contain item i.

The i^{th} of the next K lines contain k_i space separated integers, the buildings that contain item i.

The next M lines contain 3 space separated integers, a_i, b_i, and c_i, representing a two-way road between buildings a_i and b_i, taking c_i minutes to travel.

Output Specification

Output one integer, the minimum time it takes to obtain all K items in minutes.

Sample Input 1

```
4 4 3
9 10 10
1 1 1
3
4
2
1 2 3
2 3 5
2 4 4
3 4 10
```

Sample Output 1

```
20
```

Explanation for Sample Output 1

This sample case satisfies the condition of subtask 1.

The optimal way to obtain all items is:

- Create item 1 yourself.
- Go from building 1 to building 2.
- Go from building 2 to building 4. Collect item 2 here.
- Go from building 4 to building 2. Collect item 3 here.

In total, you take $9 + (3 + 4 + 4) = 20$ minutes, which is the minimum time possible.

Sample Input 2

```
5 6 2
1000000000 1000000000
1 2
3
4 5
1 2 1
1 3 4
2 3 2
2 4 1
3 5 6
5 4 2
```

Sample Output 2

```
6
```

Explanation for Sample Output 2

This sample case satisfies the condition of subtask 2.

The optimal way to obtain all items is:

- Go from building 1 to building 2.
- Go from building 2 to building 3. Collect item 1 here.
- Go from building 3 to building 2.
- Go from building 2 to building 4. Collect item 2 here.

In total, you take $1 + 2 + 2 + 1 = 6$ minutes, which is the minimum time possible.

Sample Input 3

```
4 6 3
3 3 5
2 3 1
2 3
2 3 4
3
1 2 4
1 3 10
2 3 6
1 4 2
2 4 3
3 4 8
```

Sample Output 3

```
9
```

Explanation for Sample Output 3

The optimal way to obtain all items is:

- Go from building 1 to building 2. Collect item 1 and 2 here.
- Create item 3 yourself.

In total, you take $4+5=9$ minutes, which is the minimum time possible.