

## 1 Coding Question:

### Reminders

- Must be coded individually in your choice of either Python, Java, C, C++, or C#
- Submitted through Gradescope, and there are hidden test cases
- There is a class-wide runtime leaderboard on Gradescope
- We encourage the use of Piazza for debugging help
- Please do not cheat

### Problem

Suppose you hope to dig up treasures in Moria with a team of dwarf warriors and Gandalf the Grey. Gandalf has warned you that treasures are not the only things in Moria—there are also other dangerous beings like Balrog and Orcs who would rob you. Thanks to your team, you know exactly where all mining locations (denote as  $t_i, i \in [1, n]$ , where  $n$  is the total number of mining locations) are as well as the values inside these locations (denote as  $p_i, i \in [1, n]$ , and  $p_i \in (-\infty, +\infty)$ ). If the mining location  $t_i$  contains a treasure,  $p_i \in [0, +\infty)$ , but if it has a Balrog or Orcs waiting for you,  $p_i \in (-\infty, 0)$ . Moreover, in order to reach a mining location ( $t_j$ ), it is possible that there are other mining locations that you must visit first—consider visiting this set of locations ( $T_j$ ) as the prerequisite condition of reaching  $t_j$ . Note: There is no cost traveling between mining locations, and the value of each visited mining location is only counted once (revisiting a location would not give you any more treasure or robbery).

Your goal is to select a subset of mining locations to visit so that your team can carry away maximum value of treasure from Moria.

### Input:

- Input should be read in from stdin.
- The first line will contain the total number of mining locations ( $n$ ), and the number of nodes that has at least 1 prerequisite ( $m$ )
- The second line contains the value of each mining location (in form of  $p_1 p_2 \dots p_i \dots p_n$ , where  $i \in [1, n]$ )
- Each line after will contain a list of numbers, with the first number ( $i$ , where  $i \in [1, n]$ ) being the index of a mining location ( $t_i$ ), and the rest of the numbers being  $T_i$  (the prerequisite mining locations of  $t_i$ ).

### Constraint:

- You can expect  $n \in [1, 2000]$  and  $p_i \in [-2000, 2000]$

### Output:

- The output should be written to stdout.
- The output should be your profit value

## Examples

### Example 1

input:

```
2 1
4 -3
1 2
```

output:

```
1
```

### Example 2

input:

```
6 3
100 200 150 -200 -100 -50
1 4 5
2 5
3 6
```

output:

```
200
```

### Example 3

input:

```
8 6
1 -5 -2 1 1 0 10 2
1 2
3 1 4
4 2 6
5 3 7
7 4 8
8 6
```

output:

```
9
```