[C5] Chase through the Metaverse (300 pts)

Time Limit: 1s
Memory Limit: 512MB

Problem Description

The chase is on! You stare at the back of your suspect running through the crowd of digital avatars. You follow him to the different worlds and zones in the metaverse and eventually stop in a world made up of pixelized mushrooms and pipes.

From your extensive knowledge of pop culture in the late 90's, you recognize that this zone is a remake of the popular game Mario 64. As you pause to catch your breath, you spot the suspect suddenly blink out of existence and reappear at an upper level near an elevator to the exit.

You then recall a popular speedrunning strategy in Mario 64 that makes use of bugs in the game's programming to teleport the player to the exit. Notes about the speedrunning strategy is detailed below:

The World of Mario 64

- The ground in the world of mario 64 is made up of N triangles labeled from 0 to $N-1$.
- There are E bidirectional paths between them represented by the tuple (U_i, V_i).
- Each triangle has an associated speed requirement value S_i .
- The player also has its own speed value P. This value can only be increased on the entry point ground triangle 0.

Instantaneous Travel Between Triangles

- A player can traverse two triangles only if a path exists between them.
- A player cannot traverse two triangles if his speed is zero.
- A player can traverse two triangles if his speed is equal to the speed requirement of the destination triangle. In addition to this, a bug was found that allows travel if the player speed is a multiple of the speed requirement of the destination triangle.
- A player can stay in a triangle to reduce his speed to meet the speed requirement for the next triangle. Note that there is no way to increase the player speed once its lowered. Also, this reduction happens instantaneously.
- Travel in this manner is instantaneous and appears as rapid teleportation to the destination.

Note that the speed requirement of the starting triangle 0 doesn't really matter, and for simplicity will always be set to 1.

The elevator to the next level is located at triangle $N-1$. In order to chase the NELOC agent to the next level, determine the minimum initial speed P_0 that must be built up at triangle 0.

Input Specification

Input will begin with an integer T denoting the number of test cases. T test cases follow.

Each test case begins with a line containing two space separated integers, N - the number of ground triangles and E - the number of paths between ground triangles.

The next line in a test case contains N space separated integers denoting the speed requirement for a ground triangle S_i.

E lines then follow with two space separated integers (U_i, V_i) each. These represent a bidirectional path between triangle U_i and triangle V_i. Note that we designate the first triangle as 0 and the final triangle with the elevator as $N - 1$.

Output Specification

For each test case, output an integer P_0 denoting the speed that needs to be built up at the entry point ground triangle 0 to reach the elevator at triangle $N - 1$.

Constraints

$1 \leq T \leq 10$
$1 \leq E \leq 10^4$
$2 \leq N \leq 1000$
$1 \leq S_i \leq 1000$
$S_0 = 1$
$0 \leq U_i, V_i \leq N - 1$
It is guaranteed that there is a path from triangle 0 to $N - 1$.

Sample Input

```
2
3 2
1 6 5
0 1
1 2
5 5
1 3 6 8 5
0 1
1 3
1 2
2 4
3 4
```

Sample Output

```
6
6
```

Explanation

For test case 1, the path between the triangles follows the order $0 \Rightarrow 1 \Rightarrow 2$.

Traversing in test case 1 follows the sequence of events below:

1. Build up a speed of 6 in triangle 0
2. Travel to triangle 1 (speed requirement of 6)
3. Reduce speed to 5
4. Travel to triangle 2 (speed requirement of 5)

For test case 2, the path between the triangles follows the order $0 \Rightarrow 1 \Rightarrow 2 \Rightarrow 4$.

Traversing in test case 2 follows the sequence of events below:

1. Build up a speed of 6 in triangle 0
2. Travel to triangle 1 (speed requirement of 3) by abusing the bug (6 is a multiple of 3)
3. Travel to triangle 2 (speed requirement of 6)
4. Reduce speed to 5
5. Travel to triangle 4 (speed requirement of 5)