Algorithm Design Laboratory with Applications

Prof. Stefano Leucci

Problem: *Water Treatment Plant.*

The water treatment plant of Algoland has a very complex system of pipes and valves. There are n main pipes $p_1, p_2, \dots, p_n$ that leave the plant and m valves $v_1, \dots, v_m$ that control the water flow through these pipes.

Opening the i-th valve v_i allows 1 Kiloliter (Kl) of water per second to flow into a subset of pipes $P_i \subseteq \{p_1, p_2, \dots, p_n\}$ to which v_i is connected. It is possible for a pipe p_j to receive water from more than one valve: if there are k open valves connected to p_j, then the amount L_j of water flowing into p_j will be of exactly k Kl/s.

Each pipe p_j serves a different neighborhood of Algoland which has a specific *demand* D_j of water (in Kl/s). To avoid the water pressure to rise to dangerous levels, it is critical for L_j not to exceed D_j.

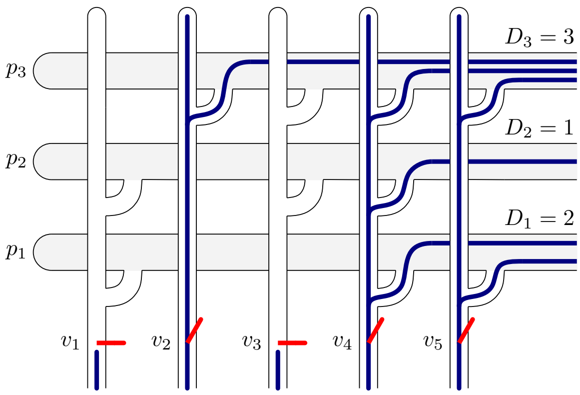
Your task is to design an algorithm that, given n, m, the subsets P_i, and the demands D_j, determines if there is a subset of valves that can be open (simultaneously) in order to meet all the pipe demands, without exceeding them.

Input. The input consists of a set of instances, or *test-cases*, of the previous problem. The first line contains the number T of test-cases. The first line of each test case contains the integers n and m. The second line contains the n integers $D_1, \dots, D_n$. The i-th next m lines contains $1 + |P_i|$ integers: the first integer is $|P_i|$, while each each of the following integers is the index j of a pipes $p_j \in P_i$.

Output. The output consists of T lines. The i-th line is the answer to the i-th test-case and is the character `Y` if there is a set of valves that can be opened to exactly satisfy all the demands, and `N` otherwise.

Assumptions. $1 \le T \le 10$; $1 \le n \le 50$; $1 \le m \le 40$; $\forall j = 1, \dots, n$, $0 \le D_j \le 40$.

Example.



Input (corresponding to the above picture):

```
1
3 5
2 1 3
2 1 2
1 3
2 2 3
3 1 2 3
2 1 3
```

Output (corresponding to valves v_2, v_4, and v_5. See also the above picture):

```
Y
```

Requirements. Your algorithm should require $O^*(2^{\frac{m}{2}})$ time (with reasonable hidden polynomial factors in n and m).

Notes. A reasonable implementation should not require more than 3 seconds for each input file.