

# Problem: Dynamic Resource Allocation at NovaCorp

## Story

In the year 2150, NovaCorp stands at the forefront of technological innovation, operating a massive orbital research complex. The complex houses  $n$  cutting-edge research projects, each requiring specific combinations of  $m$  rare cosmic resources—ranging from quantum-entangled processors to stabilized dark matter cores.

Each project has a unique resource requirement pattern. The cosmic market is highly volatile, causing project resource demands to fluctuate dramatically. As Chief Resource Allocator, you must perform real-time feasibility checks at  $q$  critical decision points.

At each moment  $t$ , you receive updated total demand values  $D_{t,i}$  for each project  $i$ —representing the total units needed across all resources that project  $i$  requires. Given the fixed resource capacities  $C_j$  aboard the station, you must determine whether a **perfect allocation** exists that exactly satisfies every project's demand without exceeding any resource's capacity.

The fate of humanity's technological advancement rests on your algorithmic prowess!

## Formal Description

You are given:

- $n$  projects and  $m$  resource types.
- A binary matrix  $A$  of size  $n \times m$ , where  $A_{i,j} = 1$  iff project  $i$  requires resource  $j$ .
- Resource capacities  $C_1, C_2, \dots, C_m$ .
- $q$  time points, each with demand array  $D_{t,1}, D_{t,2}, \dots, D_{t,n}$ .

For each time  $t$ , determine whether there exists a non-negative allocation matrix  $X^{(t)} \in \mathbb{Z}_{\geq 0}^{n \times m}$  such that:

$$\begin{aligned} \forall i \in [1, n], \quad \sum_{j=1}^m X_{i,j}^{(t)} &= D_{t,i} \\ \forall j \in [1, m], \quad \sum_{i=1}^n X_{i,j}^{(t)} &\leq C_j \\ \forall i, j, \quad A_{i,j} = 0 &\implies X_{i,j}^{(t)} = 0 \end{aligned}$$

## Input Format

The first line contains three integers  $n, m, q$  ( $1 \leq n, m \leq 50, 1 \leq q \leq 1000$ ).

The next  $n$  lines each contain a string of length  $m$  consisting of '0' and '1'. The  $i$ -th string describes which resources project  $i$  needs.

The next line contains  $m$  integers  $C_1, C_2, \dots, C_m$  ( $1 \leq C_j \leq 10^4$ ).

Finally,  $q$  lines follow. The  $t$ -th of these lines contains  $n$  integers  $D_{t,1}, D_{t,2}, \dots, D_{t,n}$  ( $0 \leq D_{t,i} \leq 10^4$ ).

## Output Format

Output  $q$  lines. The  $t$ -th line should contain YES if a feasible allocation exists at time  $t$ , otherwise NO.

## Constraints

- $1 \leq n, m \leq 50$
- $1 \leq q \leq 1000$
- $1 \leq C_j \leq 10^4$
- $0 \leq D_{t,i} \leq 10^4$
- Each project requires at least one resource (each row of  $A$  contains at least one '1').
- Time limit: 2.0 seconds
- Memory limit: 512 MB

## Sample Input

```
3 2 4
10
01
11
5 5
2 3 4
1 2 3
0 0 0
5 5 5
```

## Sample Output

```
YES
YES
YES
NO
```

## Sample Explanation

### Resource requirements:

- Project 1 needs only resource 1.
- Project 2 needs only resource 2.
- Project 3 needs both resources 1 and 2.

**Time 1:** Demands =  $[2, 3, 4]$ . One possible allocation:

- Project 1: 2 units from resource 1.
- Project 2: 3 units from resource 2.
- Project 3: 3 units from resource 1 and 1 unit from resource 2.

Total used: resource 1: 5, resource 2: 4 (within capacities 5 and 5).

**Time 2:** Demands =  $[1, 2, 3]$ . Allocation exists.

**Time 3:** Demands =  $[0, 0, 0]$ . Trivially satisfied by allocating nothing.

**Time 4:** Demands =  $[5, 5, 5]$ . Impossible because:

- Projects 1 and 3 share resource 1. Maximum they can get is  $C_1 = 5$ .
- Projects 2 and 3 share resource 2. Maximum they can get is  $C_2 = 5$ .
- But project 1 needs 5, project 2 needs 5, and project 3 needs 5 from both resources.
- Resource 1 would need at least  $5 + 5 = 10 > 5$ , similarly for resource 2.