Flow 2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

Little Cyan Fish, alongside his teammate Qingyu Xiao from Peking University, is engrossed in an algorithm class. Their current topic of interest is the paper *Maximum Flow and Minimum-Cost Flow in Almost-Linear Time*[1], and the paper *All-Pairs Max-Flow is no Harder than Single-Pair Max-Flow: Gomory-Hu Trees in Almost-Linear Time*[2]. Little Cyan Fish is particularly fascinated by the maximum flow problem and seeks your assistance in tackling a related challenge. Before diving into the problem, let's revisit some fundamental concepts from the textbook to ensure a clear understanding.

We call an undirected graph $G(V, E)$ a simple graph if and only if it does not have multiple edges or self-loops. For a simple graph $G(V, E)$ with $|V| \geq 3$, let s and t be any two distinct vertices (called **source** and **sink**, respectively). A **flow** is a map $f : E \mapsto \mathbb{R}_{\geq 0}$ that satisfies the following:

- **(Capacity constraints)**: the flow of an edge cannot exceed 1, in other words, $0 \leq f_{uv} \leq 1$ for all $(u, v) \in E$.
- **(Conservation of flows)**: The sum of the flows entering a node must equal the sum of the flows exiting that node, except for the source and the sink. In other words,

$$\forall v \in V \setminus \{s, t\} : \sum_{u:(u,v)\in E} f_{uv} = \sum_{u:(v,u)\in E} f_{vu}$$

Please note that, in this problem, the capacity of all the edges equals 1. In other words, **the edges in the given graph are all unweighted**.

The **value of flow** is the amount of flow passing from the source to the sink. Formally for a flow $f : E \mapsto \mathbb{R}_{\geq 0}$ it is given by:

$$|f| = \sum_{v:(s,v)\in E} f_{sv} = \sum_{u:(u,t)\in E} f_{ut}$$

The maximum s,t-flow, denoted by $\text{maxflow}(s, t)$, is the maximum value of all possible flows f with the source s and sink t. Specifically, we define $\text{maxflow}(u, u) = 0$ for all $u \in V$.

Now, Little Cyan Fish is doing the assignment for the algorithm class. The assignment gives Little Cyan Fish a simple undirected graph $G(V, E)$ with n vertices, labeled from 1 to n. His task is to compute the matrix $A = F(G)$, where $A_{u,v} = \text{maxflow}(u, v)$ for every $u, v \in V$. Finding this straightforward, Little Cyan Fish is more intrigued by the inverse problem: Given a matrix $A_{u,v}$, how can one construct an undirected simple graph G such that $F(G) = A$?

Of course, this problem is really hard. So Little Cyan Fish will give you some simpler matrices — the matrix consisting **only of the integers** $\{0, 1, 2, 3\}$. In this case, can you solve the challenge by Little Cyan Fish?

[1]by Li Chen, Rasmus Kyng, Yang P. Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva, in 2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS), available at arXiv:2203.00671

[2]by Amir Abboud, Jason Li, Debmalya Panigrahi, and Thatchaphol Saranurak, in 2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)

Input

There are multiple test cases in a single test file. The first line of the input contains a single integer T ($1 \le T \le 10^3$), indicating the number of test cases.

For each test case, the first line of the input contains a single integer n ($1 \le n \le 300$).

The next n lines describe the matrix A. The i-th ($1 \le i \le n$) line of these lines contains n integers $A_{i,1}, A_{i,2}, \cdots, A_{i,n}$. It is guaranteed that $0 \le A_{i,j} \le 3$ for all $1 \le i, j \le n$.

It is guaranteed that the sum of n^2 over all test cases does not exceed 9×10^6.

Output

For each test case, if there is no such graph G satisfying $F(G) = A$, print a single line "`No`".

Otherwise, the first line of the output contains a single line "`Yes`". The next line of the output contains a single integer m, indicating the number of edges you used. The next m lines should contain two integers x and y, indicating an edge. You need to make sure that your solution is a simple graph without multiple edges or self-loops.

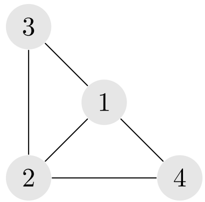
If there are multiple solutions, you may print any of them.

Example

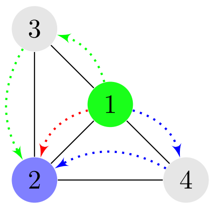
standard input	standard output
4 4 0 3 2 2 3 0 2 2 2 2 0 2 2 2 2 0 8 0 2 2 0 0 1 1 1 2 0 2 0 0 1 1 1 2 2 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 2 2 1 1 1 0 0 2 0 2 1 1 1 0 0 2 2 0 3 0 1 2 1 2 3 2 3 1 12 0 2 2 2 2 2 2 2 2 1 1 1 2 0 2 2 2 2 2 2 2 1 1 1 2 2 0 2 2 2 2 2 2 1 1 1 2 2 2 0 2 2 2 2 2 1 1 1 2 2 2 2 0 2 2 2 2 1 1 1 2 2 2 2 2 0 2 2 2 1 1 1 2 2 2 2 2 2 0 2 2 1 1 1 2 2 2 2 2 2 2 0 2 1 1 1 2 2 2 2 2 2 2 2 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0	Yes 5 1 2 3 1 3 2 4 1 4 2 Yes 8 1 2 2 3 3 1 6 7 7 8 8 6 1 6 4 5 No Yes 12 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 1 1 10 10 11 11 12

Note

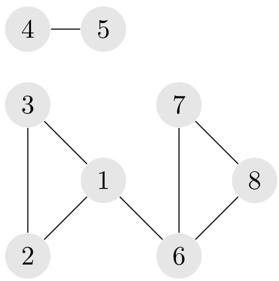
In the first test case, one possible graph is shown in the following figure.



Taking $A_{1,2} = 3$ as an example, the following figure shows that $|f_{\max}| = 3$. So the constraints for $\operatorname{maxflow}(1,2)$ is satisfied.



In the second test case, one possible graph is shown in the following figure.



In the third test case, it is obvious that $A_{u,u} = 0$ was not satisfied. Therefore, there was no possible graph corresponding to this matrix.