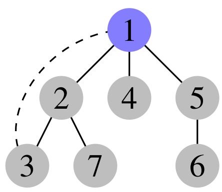
Problem
DFS Order 6
Time limit: 1 second

Little Cyan Fish loves DFS orders. Today he is studying them once more, but on undirected simple graphs rather than rooted trees.

Fix a connected undirected simple graph G on n vertices labeled 1 to n, and a starting vertex s. The *DFS order* of G from s is the sequence in which the vertices are first visited by the depth-first search shown below; ties are broken by always descending into the unvisited neighbor with the smallest label, so the DFS order is unique.

Algorithm 1 The DFS order used in this problem

```
procedure DFS(vertex x)
    Mark x as visited
    Append x to the end of dfs_order
    for each vertex y adjacent to x in G, in ascending order of label do
        if y is not yet visited then
            DFS(y)
        end if
    end for
end procedure
procedure GENERATE(G, vertex s)
    Mark all vertices as unvisited
    Let dfs_order be an empty list
    DFS(s)
    return dfs_order
end procedure
```



A graph with 7 vertices and 7 edges. The DFS order from vertex 1 is $[1, 2, 3, 7, 4, 5, 6]$.

Little Cyan Fish has prepared n permutations $a_1, a_2, \ldots, a_n$ of 1 to n. Each $a_i = [a_{i,1}, a_{i,2}, \ldots, a_{i,n}]$ is what he claims would be the DFS order from vertex i.

Reconstruct any connected undirected simple graph G on the vertices $1, 2, \ldots, n$ such that the DFS order from every starting vertex i equals a_i, or determine that no such graph exists.

Input

There are multiple test cases. The first line of the input contains a single integer T ($1 \leq T$), indicating the number of test cases.

For each test case, the first line contains a single integer n ($1 \leq n \leq 200$). Each of the next n lines contains n integers; the i-th of these lines contains $a_{i,1}, a_{i,2}, \ldots, a_{i,n}$ ($1 \leq a_{i,j} \leq n$) — the DFS order Little Cyan Fish claims is produced when the search starts from vertex i. It is guaranteed that $a_{i,1} = i$, and each row a_i is a permutation of $1, 2, \ldots, n$.

It is guaranteed that the sum of n^2 over all test cases does not exceed 4×10^4.

Output

For each test case, if no suitable graph exists, output a single line containing "`No`".

Otherwise, output "`Yes`" on the first line. On the next line, output a single integer m ($n-1 \le m \le \frac{n(n-1)}{2}$) — the number of edges in your graph.

Each of the following m lines must contain two integers u and v ($1 \le u, v \le n$, $u \ne v$), denoting an undirected edge between vertices u and v. The resulting graph must be simple and connected, and its DFS order from each vertex i must equal a_i.

If multiple graphs satisfy the requirements, output any of them.

Sample Input 1

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

Sample Output 1

```
Yes
2
1 2
2 3
No
```

Explanation of Sample 1: In the test case, the path $1-2-3$ is a valid graph: its DFS orders starting from vertices 1, 2, and 3 are $[1,2,3]$, $[2,1,3]$, and $[3,2,1]$ respectively. In the second test case, no suitable graph exists.