Problem C
Froggy and Froggo

Time limit: 8 seconds
Memory limit: 1024 megabytes

Problem Description

The king of the frogs has two sons, Froggy and Froggo. He wants to divide his kingdom into two parts, one for Froggy and one for Froggo.

The kingdom of the frogs is a square of $n \times n$ grids, with Froggy's castle at the top-left corner $(1, 1)$ and Froggo's castle at the bottom-right corner (n, n). Each grid cell has a value, and these values may be different.

The king wants to be fair, but he doesn't want an over complicated partition. He needs your help to generate a partition satisfying the following restrictions:

- The boundary of the partition is a path.
- The boundary starts at the bottom-left corner of some grid cell lying on the bottom row or the leftmost column of the grid.
- The boundary ends at the top-right corner of some grid cell lying on the top row or the rightmost column of the grid.
- The boundary consists of only edges of grid cells.
- If we walk along the boundary from its start to its end, we only go either upward or right. Moreover, we can only make *at most five* turns.
- The sum of values belongs to Froggy equals the sum of values belongs to Froggo.

Please write a program to calculate the minimum turns of a boundary satisfying all restriction above. If the minimum number of turns is greater than 5, your program should simply output `NO`.

Input Format

The first line contains an integer T indicating the number of test cases. The first line of each testcase contains an integer n. The following n lines contain n integers each, the j^{th} element in the i^{th} line a_{ij} is the value of the grid cell lying on the intersection of the i^{th} row and the j^{th} column.

Output Format

If the minimum number of turns is at most five turns, output `YES` and the minimum number of turns. Otherwise, output `NO`.

Technical Specification

- $1 \leq T \leq 20$
- $1 \leq n \leq 50$
- $1 \leq a_{ij} \leq 10^4$
- $1 \leq i, j \leq n$
- The sum of all n is at most 50.

Sample Input 1

```
2
3
3 3 2
4 2 2
2 1 1
3
5 5 5
5 1 1
1 1 16
```

Sample Output 1

```
YES 3
YES 2
```

Sample Input 2

```
1
2
15 3
3 3
```

Sample Output 2

```
NO
```

Note

The following picture is the partition corresponding to the first test case in sample input 1.

