Mid Term Exam | Introduction to Algorithms

Sort Edges

Problem Statement

You will be given E, the number of edges. For each you will be given A and B which means there is an edge between A and B.

You need to sort all edges in such a way that the edges are sorted in ascending order for A. If multiple edges have same A, then you will sort them according to B in ascending order.

Input Format

- **First line will contain E.**
- **Next E lines will contain A and B.**

Constraints

1. **1 <= E <= 10^5**
2. **0 <= A, B <= 10^9**

Output Format

- **Output all the edges in sorted order.**

Sample Input 0

```
6
3 4
3 2
3 5
1 2
1 4
1 5
```

Sample Output 0

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

Same Component

Problem Statement

You will be given a 2D matrix of size NxM which will contain only dot(.) and minus(-) where dot(.) means you can go in that cell and minus(.) means you can't.

You can move in only 4 directions (Up, Down, Left and Right).

You will be given the indexes of to cells - S(si,sj) and D(di,dj). You need to tell if these cells are in the same component or not where you can go from S to D.

Input Format

- **First line will contain N and M.**
- **Next you will be given the 2D matrix.**
- **Next line will contain si and sj.**
- **Last line will contain di and dj.**

Constraints

1. **1 <= N, M <= 1000**
2. **0 <= si, di < N**
3. **0 <= sj, dj < M**

Output Format

- Output "YES" if those cell are in the same component, "NO" otherwise.

Sample Input 0

```
5 4
..-.
---.
..-.
--..
....
0 1
3 2
```

Sample Output 0

```
NO
```

Sample Input 1

```
5 4
....
---.
..-.
--..
....
0 1
3 2
```

Sample Output 1

```
YES
```

Area of Component

Problem Statement

You will be given a 2D matrix of size **NxM** which will contain only dot(.) and minus(-) where dot(.) means you can go in that cell and minus(.) means you can't.

You can move in only 4 directions (Up, Down, Left and Right).

The area of a component is the number of dots(.) in that component that can be accessible. You need to tell the minimum area of all available components.

Note: If there are no components, print -1.

Input Format

- First line will contain **N** and **M**.
- Next you will be given the 2D matrix.

Constraints

1. 1 <= **N, M** <= 1000

Output Format

- Output the minimum area.

Sample Input 0

```
6 5
..-..
..-..
-----
.-...
.----
.....
```

Sample Output 0

```
3
```

Sample Input 1

```
3 3
---
---
---
```

Sample Output 1

```
-1
```

Can Go?

Problem Statement

You will be given **N** numbers of nodes, **E** numbers of edges in a graph. For each edge you will be given **A, B** and **W** which means there is a connection from A to B for which you need to give W cost. The value of nodes could be **from 1 to N**.

You will be given a source node **S**. Then you will be given a test case **T**, for each test case you will be given a destination node **D** and a cost **DW**. You need to tell if you can go to the destination from source using DW cost.

Input Format

- First line will contain **N** and **E**.
- Next E lines will contain **A** and **B**.
- Next line will contain source node **S**.
- Next line will contain **T**, the number of test cases.
- For each test case, you will get **D** and **DW**.

Constraints

1. 1 <= **N** <= 1000
2. 1 <= **E** <= N*(N-1)
3. 1 <= **S** <= N
4. 1 <= **T** <= 1000
5. 1 <= **D** <= N
6. 0 <= **DW** <= 10^9

Output Format

- Output "**YES**" or "**NO**" for each test case if it is possible to go from S to D in DW cost.

Sample Input 0

```
5 7
1 2 10
1 3 2
3 2 1
2 4 7
3 4 2
```

```
4 5 5
2 5 2
1
5
1 0
2 5
3 1
4 4
5 6
```

Sample Output 0

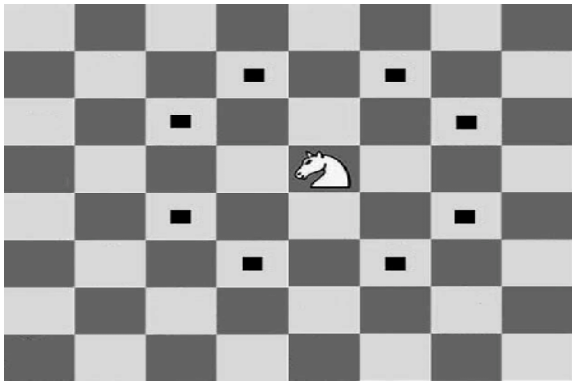
```
YES
YES
NO
YES

YES
```

Knight Moves

Problem Statement

You will be given a chessboard of **NxM** size. You can move anywhere in the chessboard freely. You will be given two cells - the knight's cell **K**(Ki and Kj), and the queen's cell **Q**(Qi and Qj). You need to tell the minimum number of steps for the knight to attack the queen if the queen doesn't move. A knight move in 8 directions. The directions are given below:



Input Format

- First line will contain **T**, the number of test cases.
- First line of each test case will contain **N** and **M**.
- Second line of each test case will contain **Ki** and **Kj**.
- Third line of each test case will contain **Qi** and **Qj**.

Constraints

1. 1 <= **T** <= 100
2. 1 <= **N, M** <= 100
3. 0 <= **Ki, Qi** < N
4. 0 <= **Kj, Qj** < M

Output Format

- Output the minimum number of steps for the knight to reach the queen. If you can't reach to queen, print **-1**.

Sample Input 0

```
4
8 8
0 0
7 7
5 6
0 1
0 1
4 4
0 0
0 1
2 2
0 0
0 1
```

Sample Output 0

```
6
0
3
-1
```

Explanation 0

For the first test case, one of the possible answer could be this way: