ETH Zurich
Institut für Theoretische Informatik
Prof. Dr. Angelika Steger
Prof. Dr. Emo Welzl
Prof. Dr. Peter Widmayer

Algorithms Lab

Exercise – *Next Path*

Having graduated from Warthogs School of Witchcraft and Wizardry, young, promising Apprentice Arthur decides to apply for a PhD position with powerful Master Merlin. Obviously this is not so simple and Merlin accepts only those candidates that impress him with something extraordinary. Since Arthur just finished reading "Introduction to Algorithms" he proposes: "Give me a graph and two of its vertices, and I will find a shortest path between them in no time". Unfortunately, Merlin took part in Algolab course at ETH back in 2008 and he is not impressed at all. Instead, he gives Arthur a different problem: "Given a graph and two vertices, find a *second* shortest path between them in no time". You were hired to help Arthur with this task.

By a *graph* we mean a directed graph, possibly with loops but without multiple edges. A *path* in a graph G is a finite non-empty sequence of vertices of G

$$P = v_0, v_1, \ldots, v_k$$

such that (v_{i-1}, v_i) is an edge in G for each $i \in \{1, \ldots, k\}$. We call $k \geq 0$ the *length* of P and denote it $\mathrm{len}(P)$. We also say that a path P goes from v_0 to v_k. We consider two paths different if the appropriate sequences are different.

The second shortest path from s to t in G is a path P in G such that:

- There exists a path Q from s to t such that $Q \neq P$ and $\mathrm{len}(Q) \leq \mathrm{len}(P)$.
- For each path R from s to t such that $R \neq Q$ we have $\mathrm{len}(P) \leq \mathrm{len}(R)$.

Note that a second shortest path is not necessarily unique.

Given a graph G and vertices s and t, find the length of a second shortest path from s to t in G.

Input The first line of the input contains the number $t \leq 50$ of test cases. Each of the t test cases is described as follows.

- It starts with a line that contains two space separated numbers $1 \leq n \leq 100$ and $0 \leq m \leq 10000$ denoting numbers of vertices and edges of G.
- The next line contains two integers $1 \leq s, t \leq n$.
- m lines follow. Each of them contains two single-space separated numbers $1 \leq s_i, t_i \leq n$ denoting the source and the target of the i-th edge of G. No (ordered) pair of vertices appears more than once in the same testcase.

Output For every testcase you should output a single line with the length of a second shortest path from s to t in G. If such a path does not exist, output `no` in a single line.

Points There are three groups of test sets, worth 100 points in total.

1. For the first group of test sets, worth 40 points, you may assume that all given graphs are acyclic.
2. For the second group of test sets, worth 40 points, there are no additional assumptions.
3. For the third (hidden) group of test sets, worth 20 points, there are no additional assumptions.

Sample Input

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

Sample Output

```
3
no
```

Note: Put `std::ios_base::sync_with_stdio(false)` as the first line of the main procedure for faster I/O.