

---

## ACM Programming Challenges Lab

---

### Exercise 1 – Navigation

Jon has a brilliant business idea: he figured that all the navigation apps on the market only provide you with the shortest route from the start point to the destination, however there is no app that displays all the possible routes! Unfortunately, Jon cannot code it himself so he hires you to do the job.

You are given a map where crossings are numbered by nonnegative integers and streets are pairs of crossings (you can use every street in both directions). Moreover, you get a start crossing and a destination crossing. Your task is to output all the possible routes from the start to the destination without visiting a crossing more than once (i.e., all paths).

**Input** The first line of the input contains the number of test cases  $t \leq 20$ . Every test case starts with a line containing the number of crossings  $1 \leq n \leq 30$ , the number of streets  $0 \leq m \leq 435$ , the start point  $s$ , and the destination  $t$  with  $s \neq t$  and  $0 \leq s, t \leq n - 1$ . The following  $m$  lines contain two numbers  $a$  and  $b$  with  $0 \leq a < b \leq n - 1$ , indicating that the crossings  $a$  and  $b$  are connected by a street.

In the small test set the number of crossings is at most 10 and in the large test set the number of paths is bounded by 5000.

**Output** For every test case output for each path from  $s$  to  $t$  a line consisting of the crossings along the path, separated by spaces. The routes should be ordered lexicographically. Then output the total number of  $s$  to  $t$  paths on the next line.

#### Sample Input

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

#### Sample Output

```
0 1 2
1
0 1 2 3
0 2 3
2
3 0 1 4
3 2 4
3 4
3
```