Tree and Permutation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Given an integer n, an undirected tree with n nodes, and two distinct nodes s, t on the tree where each edge has a length of 1. Nodes are numbered with integers from 1 to n. Let $\operatorname{dist}(u, v)$ denote the distance between nodes u and v (i.e., the number of edges on the simple path between them). You are required to find a permutation $p_1, p_2, \cdots, p_n$ of numbers from 1 to n that satisfies the following two conditions:

- $p_1 = s, p_n = t$;
- For $d_i = \operatorname{dist}(p_i, p_{i+1})$ where $1 \le i \le n-1$, the permutation should minimize $\oplus_{i=1}^{n-1} d_i$, where $\oplus$ denotes the bitwise XOR operation.

If there are multiple permutations that satisfy the conditions, output any one of them.

Input

This problem has multiple test cases. The first line inputs a positive integer T ($T \ge 1$) indicating the number of test cases.

For each test case, the first line inputs three positive integers n, s, t ($2 \le n \le 5 \times 10^4$, $1 \le s, t \le n$, $s \ne t$). The following $n-1$ lines each contain two positive integers u, v ($1 \le u, v \le n$, $u \ne v$), indicating that there is a direct undirected road connection (i.e., an edge on the tree) between locations u and v.

It is guaranteed that the sum of n over all test cases does not exceed 5×10^5

Output

For each test case, output a line with n positive integers $p_1, p_2, \cdots, p_n$, ensuring it is a permutation of 1 to n with $p_1 = s, p_n = t$, and $\oplus_{i=1}^{n-1} d_i$ is minimized.

Examples

standard input	standard output
3 3 1 3 1 2 2 3 4 3 4 1 2 2 3 2 4 5 1 2 1 2 1 3 2 4 3 5	1 2 3 3 2 1 4 1 5 3 4 2
3 10 2 3 7 5 6 1 9 1 4 5 3 10 5 1 10 9 1 2 8 3 10 3 7 5 6 4 8 9 1 6 3 7 3 2 5 10 1 8 9 1 6 10 10 4 5 10 1 4 4 5 6 1 9 6 2 10 8 1 3 6 7 4	2 6 5 4 7 1 9 8 10 3 3 5 2 1 4 8 9 10 6 7 10 2 5 7 1 8 6 3 9 4