

## **Problem 3 : Risk**

### **Problem Link:**

[https://onlinejudge.org/index.php?option=com\\_onlinejudge&Itemid=8&page=show\\_problem&problem=508](https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=508)

### **1. Problem Statement**

You are given the map of a *Risk*-like board game consisting of 20 countries, numbered 1 to 20.

Each test set describes all borders between these countries. A border is bidirectional, and the input format lists only borders where  $I < J$ , so each border appears exactly once.

For each test set:

- Lines 1 through 19 describe borders.  
Line  $i$  begins with an integer  $X$ , meaning country  $i$  borders  $X$  countries numbered greater than  $i$ .  
The next  $X$  integers list those neighboring countries.
- Line 20 contains an integer  $N$ : the number of conquest queries.
- The next  $N$  lines each contain integers  $A\ B$ , representing a starting country  $A$  and a destination country  $B$ .

Your task:

For each query  $(A, B)$ , compute the minimum number of countries that must be conquered, including the destination country, when traveling from  $A$  to  $B$ .

This is equivalent to the shortest path length between  $A$  and  $B$  in the graph.

### **2. Hint**

Treat the map as a small, fixed-size unweighted graph with 20 nodes.

The task for each query is equivalent to finding the minimum number of edges between two nodes.

Since all edges have equal weight, Breadth-First Search (BFS) gives the shortest path distance efficiently and reliably.

Only one BFS per query is needed because the graph is tiny.

### **3. Solution Approach**

#### **1. Graph Construction**

- Read 19 lines describing higher-numbered neighbors.
- For each listed connection  $(i, v)$ , insert edges in both directions into the adjacency list.
- This produces a complete undirected graph of the board.

## 2. Shortest-Path Computation

- For each query (A, B):
  - Initialize a distance array with -1.
  - Run BFS from A, updating distances as nodes are reached.
  - Once B is assigned a distance, stop the BFS.
  - The distance value equals the minimum number of countries that must be conquered.

## 3. Output Formatting

- Follow the exact output format:  
A to B: distance
- Align numbers using fixed-width formatting rules as in the sample output.
- Print a blank line after finishing all queries in a test set.

## Pseudocode

```
testSet = 1
while (there is input):
    create graph G with 21 empty lists
    for (i = 1; i <= 19; i++):
        read X
        for (k = 1; k <= X; k++):
            read v
            G[i].push_back(v)
            G[v].push_back(i)
    read N
    cout << "Test Set #" << testSet << endl
    testSet++
    for (q = 1; q <= N; q++):
        read A, B
        dist = BFS(A, G)
        cout << setw(2) << A << " to " << setw(2) << B << ": " << dist[B] << endl
    cout << endl
```