RISK

### Input:

### Output:

### Ideas:

Tìm đường đi ngắn nhất từ I -> J => tìm đường đi giữa nhiều cặp.

Không có trọng số -> trọng số bằng 1.

Cạnh 2 chiều.

BFS -> O(T \* V \* (V+E))

Dijkstra -> O(T \* V \* (Vlog(V+E))

Bellman Ford -> O(T\* V \* (V \* E))

\*\* Floyd -> O(T \* V \* V \* V))

### Code:

function main() {

// Scanner sc = new Scanner(System.in);

// while(sc.hasNextInt())

// try {

// } catch (EOFError)

int x;

int idTest = 0;

while(cin >> x) {

idTest++;

for (int i: 1->x) {

int u;

cin >> u;

dist[1][u] = dist[u][1] = 1;

}

for(int i: 2 -> 19) {

cin >> x;

for(j: 1 -> x) {

cin >> u;

dist[i][u] = dist[u][i] = 1;

}

}

for(k: 1->20)

for(u: 1->20)

for(v: 1->20) {

dist[u][v] = min(dist[u][v], dist[u][k] + dist[k][v]);

}

int q;

cout << “Test set #” << idTest << “\n”;

cin >> q;

for(i: 1->q) {

int u, v;

cin >> u >> v;

cout << dist[u][v] << “\n”;

}

cout << “\n”;

}

}