USA Tutor

Mootube Silver

Analysis by David Yang

Problem Summary

You’re given a Tree with N<=5000 Nodes

You will have N-1 edges, which are of the form “a b c” where a and b are connected with relevance c.

Answer Q<=5000 Queries

What will we need?

We obviously need DFS, but how do we even use DFS in this problem?

These edges have a “relevance” to them.

When we query, we must follow the minimum relevance rule for any edge.

Taking into account the Min Relevance

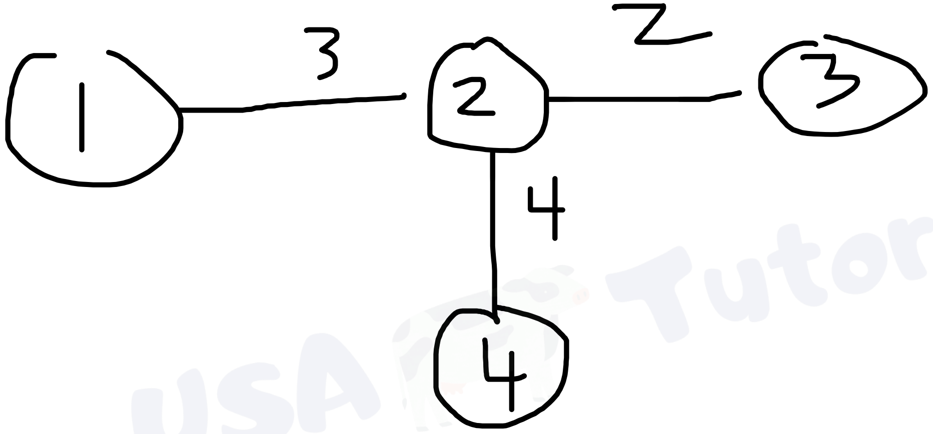
We will only ever traverse down in our DFS if and only if:

We are connected (A and B have an edge)

This edge has a relevance R >= our min relevance

Sample Case

A	B	R
1	2	3
2	3	2
2	4	4



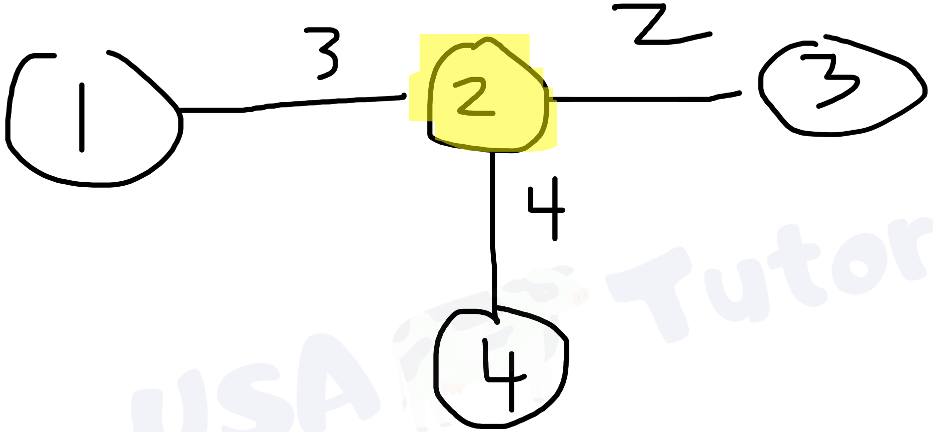
Sample Case

Queries:

1 2

4 1

3 1



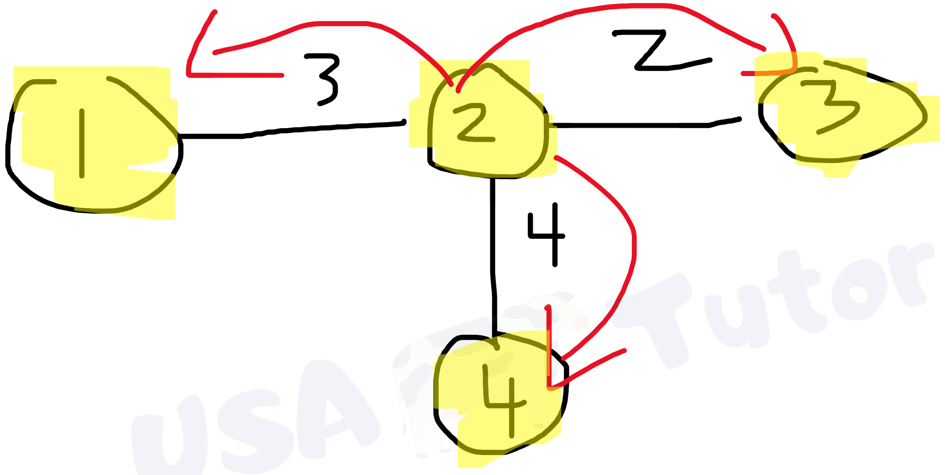
Sample Case

Queries:

1 2

4 1

3 1



3

Sample Case

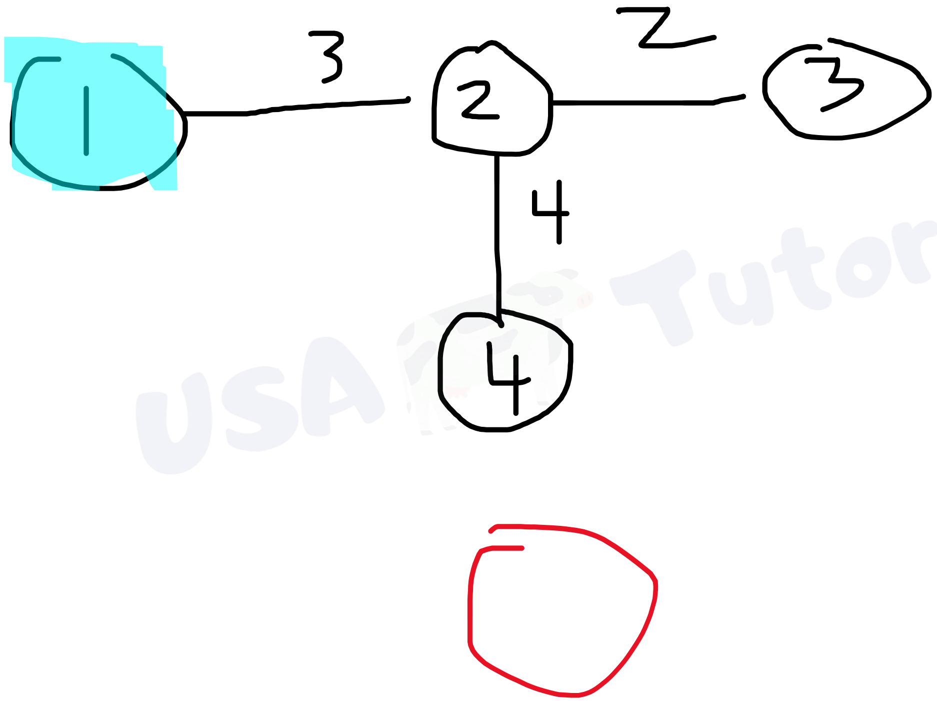
no videos

Queries:

1 2

4 1

3 1



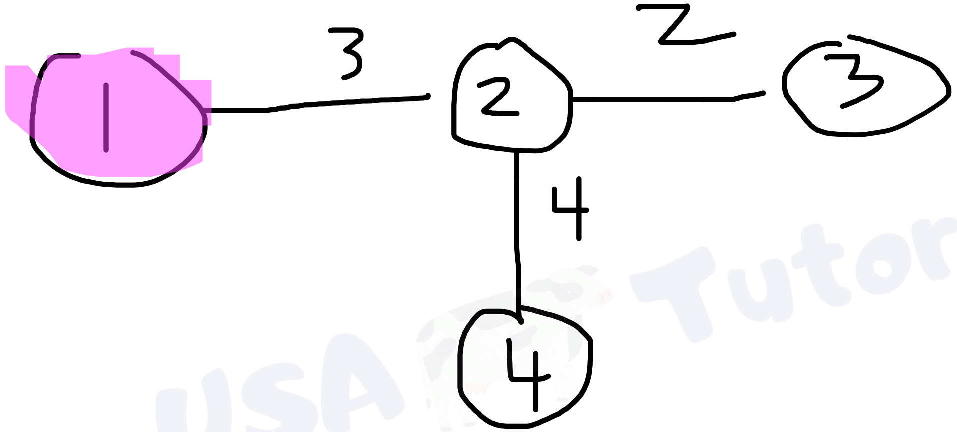
Sample Case

Queries:

1 2

4 1

3 1



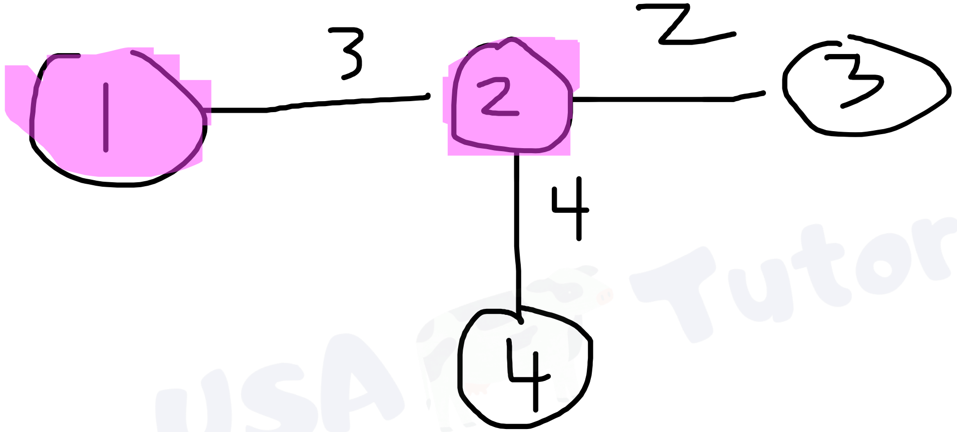
Sample Case

Queries:

1 2

4 1

3 1

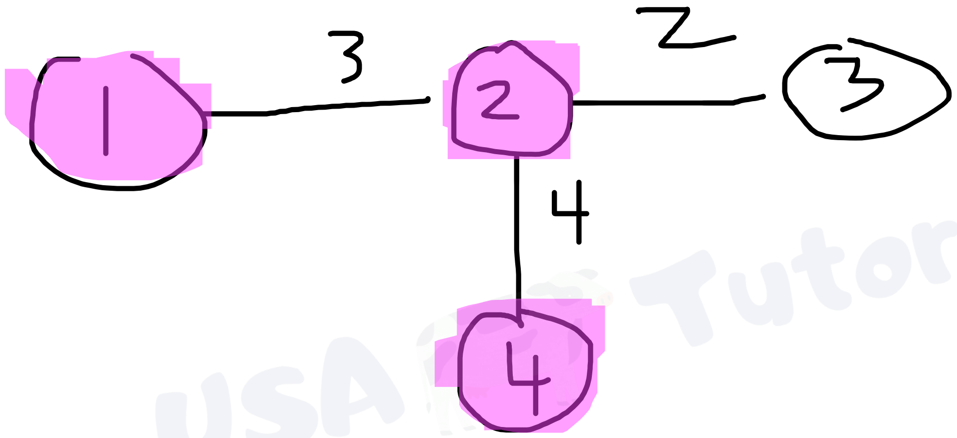


Sample Case

Queries:
1 2

4 1

3 1



2

How to Modify DFS?

Store a Pair as the object for your Adjacency List. ArrayList<Pair> adj[];
We can only pass through this edge if curWeight >= minWeight.

```
Vector<int> adj[]
Vector<pair<int,int>> adj[]
Adj[i][j] stores a pair <node, relevance> or <relevance, node>
// using r, n notation
For(int next: adj[cur]){
Int r = next.first;
If(r< K) continue;
}
```