

# ***Protecting Roads - Editorial***

## **Difficulty:**

Medium - Hard

## **Prerequisites:**

Trees - ([Tutorial](#))

Lowest Common Ancestor - ([Tutorial](#))

Minimum Spanning Tree - ([Tutorial](#))

## **Problem in Brief:**

Given a tree with  $N$  nodes and  $Q$  queries consisting of a variable( $K$ ) number of nodes, find the minimum spanning tree that spans those  $K$  nodes for each query and output the sum of all the edges in the spanning tree.

## **Editorial:**

To solve this question, we will try to solve each query independently.

Consider a single query, We will build the auxiliary tree for that query. When we have a query with  $K$  special nodes, we can construct an auxiliary tree of those  $K$  nodes and calculate the sum of edges in the auxiliary tree.

To construct an auxiliary tree, we need all  $K$  special nodes and LCAs of every pair of special nodes. But LCAs of every pair of special nodes is equal to LCAs of every adjacent pair of special nodes when the special

# ***Protecting Roads - Editorial***

nodes are sorted in preorder of the tree(arrival time of nodes in DFS). Now for the weights of the edges in the auxiliary tree, weight of edge between node  $u$  and  $v$  will be equal to distance between nodes  $u$  and  $v$  in the original tree. This can be handled if distance between every node  $u$  and root of original tree is precomputed.

Finally we just add the cost of all edges and output the answer.

## **Time Complexity:**

At each query we will sort the nodes using their preorder time. Construction of auxiliary tree is linear excluding the sorting part. Hence the Time Complexity is

$$O((N + Q) * \log(N))$$

## **Similar Problems:**

[First](#)

[Second](#)