Centroid Decomposition

Karthik Bhargav, Arnav Jain

February 2023

1 Introduction

Centroid decomposition is a fairly advanced algorithm that appears sometimes, mostly in the USACO Platinum division. It is a divide-and-conquer algorithm that allows us to break up a tree and do a variety of path computations efficiently, such as:

- Counting the number of paths of a certain length in a tree
- Counting the paths where the XOR of all the nodes is equal to K
- Finding the Minimum Number of Edges in a Path of at Least a Certain Length

2 Algorithm

2.1 What is a Centroid?

A centroid is a node in a tree of size N, where if we root the tree at that node, no subtree has more than $\frac{N}{2}$ nodes. Each tree can have at most 2 centroids.

2.2 Finding the Centroid

To find the centroid, can run a depth-first search on the tree. For each node, we can calculate the size of the subtrees (including the one of the parent to that node, via subtraction). If none of the subtrees have size greater than $\frac{N}{2}$, then we have found our centroid.

2.3 Centroid Decomposition

Once we find the Centroid of the tree, we root the tree at that node and split the tree into its subtrees. We then find the Centroid of each subtree and split it again. We repeat this process until we can no longer split our trees.

At the end of this process, we are left with a forest of $O(log(N))$ decomposed trees. This is because decomposing the tree at the Centroid ensures that the resulting subtrees have size of at most $\frac{N}{2}$, so a tree of size N will only undergo at most $O(log(N))$ splits. Since we need to do an $O(N)$ DFS for every tree, the overall time complexity of Centroid Decomposition is $O(Nlog(N))$.

By itself, Centroid decomposition does nothing. However, for each subtree we only have to check paths going through the root. For instance, if we were looking for the number of paths of length K in the entire tree, Centroid decomposition simplifies the problem so that we only need to find the number of paths going through the root of a decomposed subtree, albeit at the cost of adding an extra log factor to our time complexity.

Figure 1: Tree Rooted At Centroid

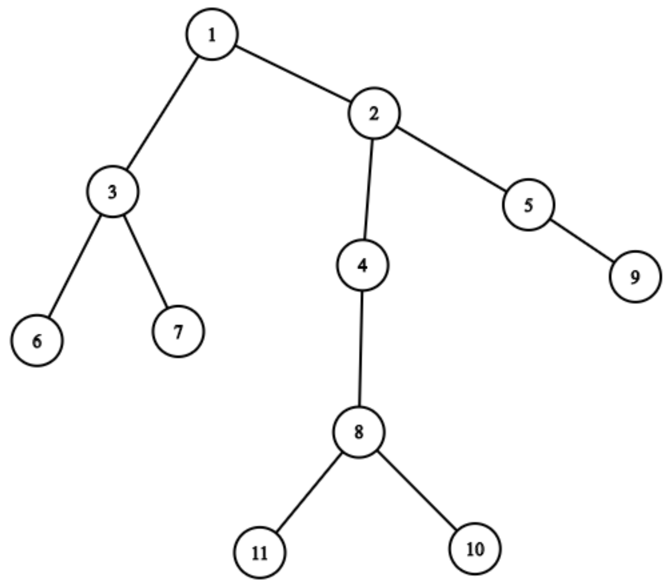
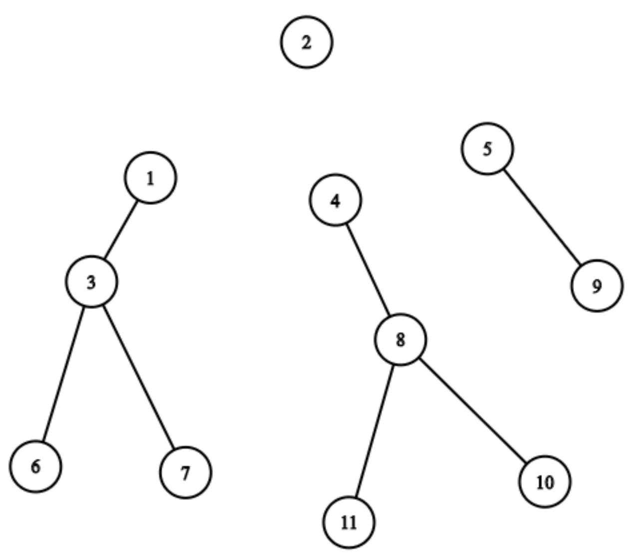


Figure 2: Tree After Splitting



3 Sample Problem

3.1 Problem Statement

Ciel has become commander of Tree Land. Tree Land has n cities connected by $n-1, n \leq 10^5$ undirected roads, and for any two cities there always exists a path between them.

Ciel needs to assign an officer to each city. Each officer has a rank — a letter from 'A' to 'Z'. So there will be 26 different ranks, and 'A' is the topmost, so 'Z' is the bottommost.

There are enough officers of each rank. But there is a special rule must obey: if x and y are two distinct cities and their officers have the same rank, then on the simple path between x and y there must be a city z that has an officer with higher rank. The rule guarantee that a communications between same rank officers will be monitored by higher rank officer.

Help Ciel to make a valid plan, or if it's impossible, output "Impossible!".

3.2 Solution

First of all, we should notice that it is never impossible for it to be impossible for Ciel to create a valid plan. The most "difficult" scenario for Ciel would be a straight chain of 10^5 cities. If we start trying arrangements of officers, such as ZYZ and ZYZXZYZ, we'll notice that for each new letter type we add, the maximum chain length increases exponentially. Since 10^5 is quite a bit smaller than 2^26, Ciel can always find an optimal arrangement, so all we have to do is figure out what the optimal arrangement is.

We'll also notice that each officer essentially splits a segment of lower ranked officers in half. This sounds awfully similar to our Centroid decomposition algorithm. We can solve the problem using Centroid decomposition. We set the Centroid of the original tree to be an officer of rank A, then the centroids of the subtrees of the original tree to be rank B, and so on. This guarantees that every path that passes through the root of a given subtree has an officer of higher rank between two lower ranked officers.
Since centroid decompositon is $O(NlogN)$, our solution runs in the same time.

4 Example Problems

USACO US Open 2013, Gold, Yin Yang

USACO February 2018, Plat, New Barns

USACO January 2018, Plat, Cow at Large