## Centroid Decomposition:

```cpp
//one based indexed
struct CentroidDecomposition {
    vector<vector<int>> adj;
    vector<int> par , subtree , visited;

    CentroidDecomposition(vector<vector<int>> &adj) : adj(adj) {
        int n = adj.size()+1;
        par.resize(n);
        subtree.resize(n);
        visited.resize(n);
        build(1, -1);
    }

    void build(int u, int p) {
        int n = treeSize(u, p);
        int centroid = getCentroid(u, p, n);
        par[centroid] = p;
        visited[centroid] = 1;

        for (auto v : adj[centroid]){
            if(!visited[v])
                build(v, centroid);
        }
    }

    int treeSize(int u, int p) {
        subtree[u] = 1;

        for (auto v : adj[u]){
            if (v != p && !visited[v])
                subtree[u] += treeSize(v, u);
        }

        return subtree[u];
    }

    int getCentroid(int u, int p, int n) {
        for (auto v : adj[u]){
            if (v != p && subtree[v] > n/2 && !visited[v])
                return getCentroid(v, u, n);
        }

        return u;
    }
};
```