

157 **LCA by using DSU:**

```
158
159 // code of DSU
160 // don not forget to fill the query array;
161
162 // vector<vector<int>> adj , query;
163 // vector<int> visited , ancestor;
164
165 // void dfs(int u){
166 //     visited[u] = 1; ancestor[u] = u;
167 //     for(auto v : adj[u]){
168 //         if(!visited[v]){
169 //             dfs(v);
170 //             g.merge(u , v); //DSU object
171 //             ancestor[g.find(u)] = u;
172 //         }
173 //     }
174 //     for(auto i : query[u]){
175 //         if(visited[i])
176 //             cout<<ancestor[g.find(i)]<<endl;
177 //     }
178 // }
179
180 // void LCA(){
181 //     ancestor.assign(n+1);
182 //     visited.assign(n+1 , 0);
183 //     dfs(root);
184 // }
```

187 **LCA in complexity $O(n)$:**

```
188 vector<int> depth(100100) , parent(100100);
189
190 int solve(int x , int y){
191     while(depth[x] > depth[y]){
192         x = parent[x];
193     }
194     while(depth[x] < depth[y]){
195         y = parent[y];
196     }
197     while(x≠y){
198         x = parent[x];
199         y = parent[y];
200     }
201     return x;
202 }
```