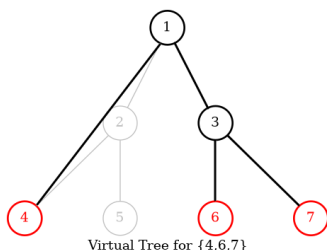


虚树

Virtual Tree

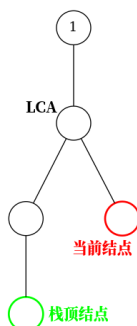
Idea: 虚树将特定的 k 个点及它们的 lca 取出建成一个新的树。



构建方法：将指定 k 个点按照 dfs 序排序，依次插入到虚树中，这里先插入树根节点以方便后续操作。插入时，维护一个栈，类似于 dfs 时的系统栈，也即栈内元素连起来是树上的一条路径（当然这里不一定连续）。

考虑如何加入一个点 x ，设 $z = \text{lca}(x, \text{stk.top}())$ ，分类讨论：

- 若 $z = \text{stk.top}()$ ，说明 x 在 $\text{stk.top}()$ 的子树中，直接将 x 入栈；
- 否则， x 不在 $\text{stk.top}()$ 的子树中，这时一直弹栈，直到遇到 z （ z 本身在栈中时）或 z 的祖先（ z 本身不在栈中时，这时把 z 入栈），然后把 x 入栈。



我们在弹栈时将弹出的元素和新的栈顶元素连边，如此便构造出了虚树。所以，所有点插入完成后记得要全部弹出栈。

构建出虚树后就可以进行后续操作了，通常是树形 dp。

Complexity: $O(k \lg n)$ 构建（采用倍增求 lca）。

Code:

```
1 namespace VT{ // Virtual Tree
2     vector<int> edge[N];
3     stack<int> stk;
4     vector<int> ver; // vertices in the virtual tree
5     void build(vector<pii> &sv){
6         // sv stores (dfn[x], x) pairs
7         // among which x are special vertices
8         sort(sv.begin(), sv.end());
9         ver.clear();
10        stk.push(1), ver.pb(1);
11        for(auto &v : sv){
12            int x = v.second;
13            if(x == 1) continue;
14            int z = LCA::lca(x, stk.top());
15            if(z == stk.top()) stk.push(x), ver.pb(x);
16            else{
17                while(!stk.empty() && LCA::dep[stk.top()] > LCA::dep[z]){
18                    int t = stk.top(); stk.pop();
```

```

19         if(!stk.empty()){
20             if(LCA::dep[stk.top()] >= LCA::dep[z])
21                 edge[t].pb(stk.top()), edge[stk.top()].pb(t);
22             else
23                 edge[t].pb(z), edge[z].pb(t);
24         }
25         if(stk.empty() || z != stk.top())    stk.push(z), ver.pb(z);
26         stk.push(x), ver.pb(x);
27     }
28 }
29 while(!stk.empty()){
30     int t = stk.top(); stk.pop();
31     if(!stk.empty())
32         edge[t].pb(stk.top()), edge[stk.top()].pb(t);
33 }
34 }
35
36 int solve(){
37     // ... solve on virtual tree
38 }
39
40 void clear(){
41     for(auto &v : ver){
42         edge[v].clear();
43         // ... clear other info
44     }
45 }
46 }
47
48 int main(){
49     // ... input
50     LCA::dfs(1, 0, 1);
51     LCA::init();
52     int q; for(scanf("%d", &q); q; q--){
53         int k; scanf("%d", &k);
54         vector<pii> sv; // special vertices
55         while(k--){
56             int a; scanf("%d", &a);
57             sv.pb(mp(LCA::dfn[a], a));
58             tag[a] = true;
59         }
60
61         VT::build(sv);
62         printf("%d\n", VT::solve());
63
64         VT::clear();
65         for(auto &v : sv)    tag[v.second] = false;
66     }
67     return 0;
68 }

```