

太戈编程
etiger.vip

信奥算法

1825

思路形成

暴力

查询：节点 u 的第 k 代祖先
循环 k 次迭代访问父节点

对于随机数据
单次查询平均复杂度 $O(\log n)$
因为随机树高度平均为 $O(\log n)$

```
6 void dfs(int u,int fa){
7     p[u]=fa;
8     d[u]=d[fa]+1;
9     for(int i=0;i<to[u].size();++i)
10         if(to[u][i]!=fa) dfs(to[u][i],u);
11 }
```

查询节点u的第k代祖先

```
12 int LA(int u,int k){
13     if(d[u]<=k) return 0;
14     for(int i=1;i<=k;++i){
15         

|  |
|--|
|  |
|  |


16     }
17     return u;
18 }
19 }
```

思路形成

暴力

对于随机数据
单次查询平均复杂度 $O(\log n)$

简化
链状

链状用一维数组储存节点序列
数组能快速索引

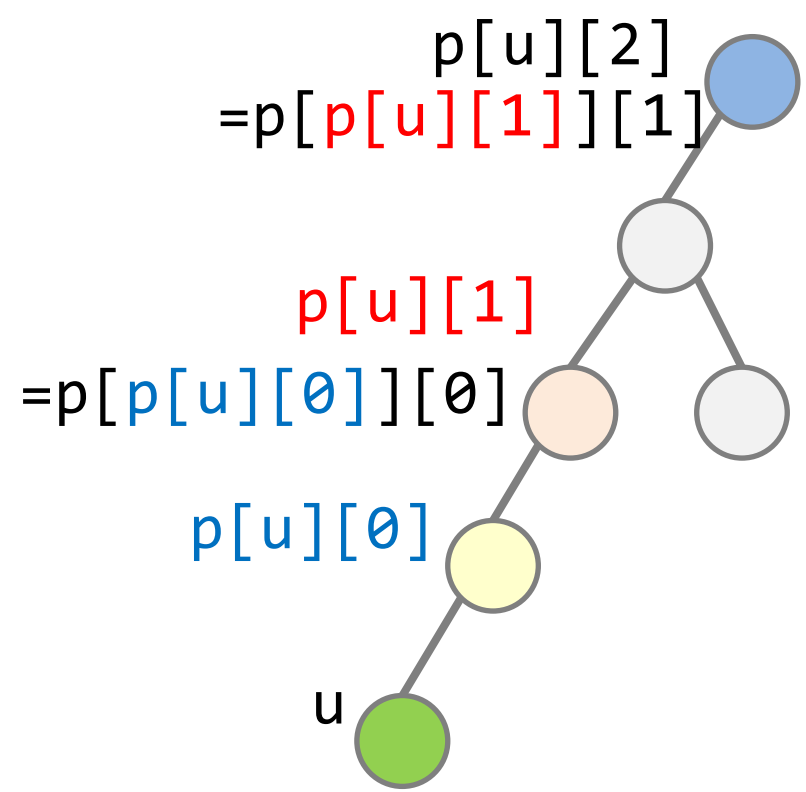
树链剖分

倍增

binary
lifting

预处理ST表(稀疏表)
 $p[u][i]$ 代表 u 的第 2^i 个祖先

```
22 cin>>n;
23 for(int i=1;i<n;++i){
24     int u,v;
25     cin>>u>>v;
26     to[u].push_back(v);
27     to[v].push_back(u);
28 }
29
30 L=log(n)/log(2)+1;
31 dfs(1,0);
32
33 long long ans=0;
34 cin>>q;
35 for(int i=1;i<=q;++i){
36     int x,k;
37     cin>>x>>k;
38     ans+=LA(x,k);
39 }
40 cout<<ans<<endl;
```

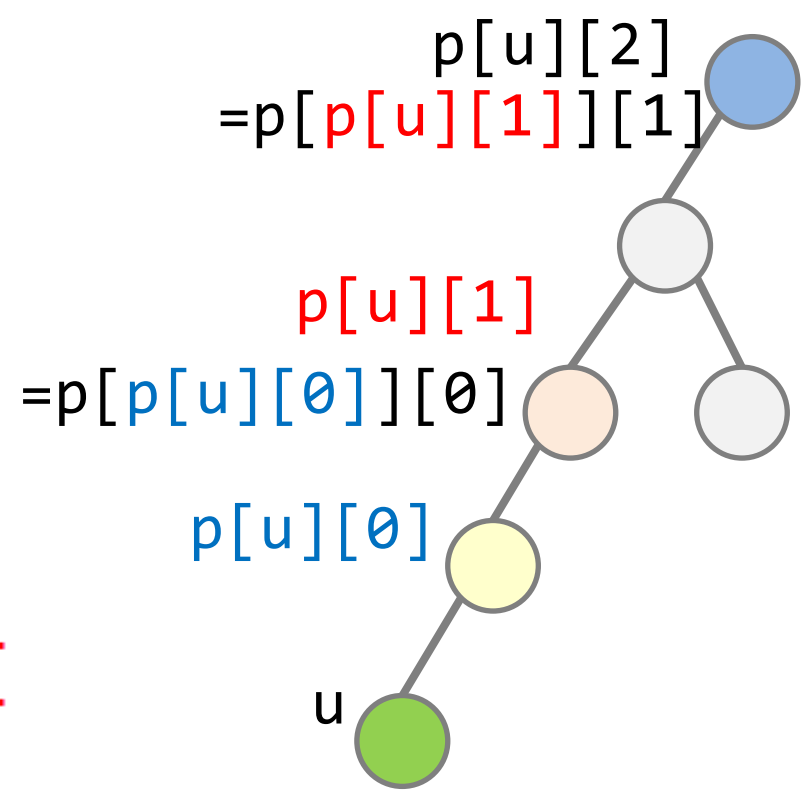


请写出 $p[u][i]$ 的递推式

```

3  const int N=200009;
4  vector<int> to[N];
5  int L,n,q,p[N][20];
6  void dfs(int u,int fa){
7      p[u][0]=fa;
8      for(int i=1;i<=L;++i)
9          p[u][i]=p[p[u][i-1]][i-1];
10     for(int i=0;i<to[u].size();++i)
11         if(to[u][i]!=fa) dfs(to[u][i],u);
12 }

```




```

13 int LA(int u, int k){
14     for(int i=0; i<=L; ++i)
15         if((1<<i)&k)
16             u=p[u][i];
17     return u;
18 }

```

$$k = (7)_{10} = (111)_2$$

$$(1 \ll 0) \& k \neq 0$$

$$(1 \ll 1) \& k \neq 0$$

$$(1 \ll 2) \& k \neq 0$$

1826

经典LCA问题

暴力

对于随机数据
单次查询平均复杂度 $O(\log n)$

目标

对于最差情况
单次查询复杂度 $O(\log n)$

倍增

binary
lifting

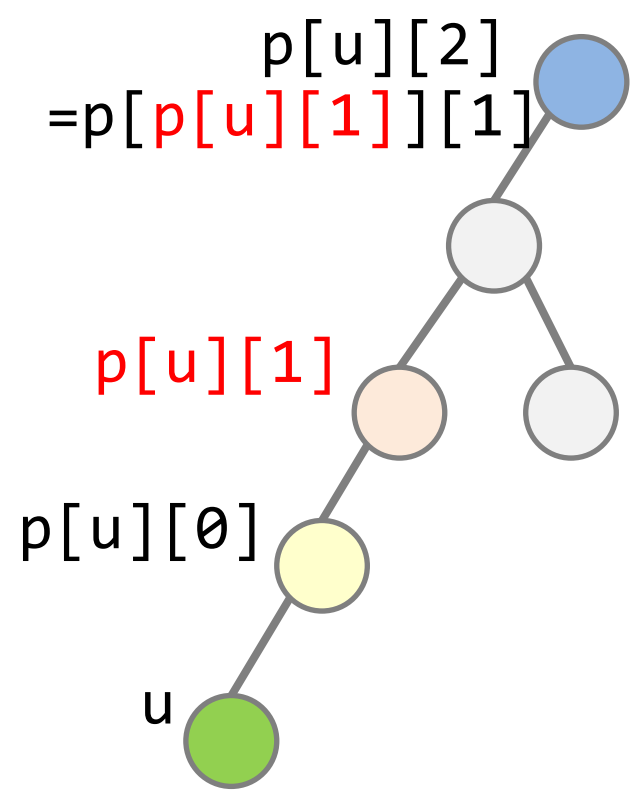
预处理ST表(稀疏表)
 $p[u][i]$ 代表 u 的第 2^i 个祖先

二分枚举答案不断逼近
需要判断两个节点的祖孙关系

```

5  int L,n,q,p[N][20];
6
7  void dfs(int u,int fa){
8
9      p[u][0]=fa;
10     for(int i=1;i<=L;++i)
11         p[u][i]=p[p[u][i-1]][i-1];
12     for(int i=0;i<to[u].size();++i)
13         if(to[u][i]!=fa) dfs(to[u][i],u);
14
15 }

```



```
17 bool up(int u, int v){  
18     return !u || tI[u] <= tI[v] && tO[v] <= tO[u];  
19 }
```

```
19 int lca(int u, int v){  
20     if(up(u, v)) return u;  
21     if(up(v, u)) return v;  
22     for(int i=L; i>=0; --i)  
23         if(!up(p[u][i], v))  
24             u=p[u][i];  
25     return p[u][0];  
26 }
```

请写出up()函数
注意有易错点

易错点

虚拟节点0号特殊处理

857

树链点权最值问题

暴力

对于随机数据
单次查询平均复杂度 $O(\log n)$

目标

对于最差情况
单次查询复杂度 $O(\log n)$

倍增

binary
lifting

预处理ST表(稀疏表)
 $p[u][i]$ 代表 u 的第 2^i 个祖先
 $st[u][i]$ 代表从 u 开始往上爬
共 2^i 个节点的点权最值

```
27 cin>>n>>m;
28 for(int u=2;u<=n;++u){
29     cin>>p[u][0];
30     son[p[u][0]].push_back(u);
31 }
32 for(int u=1;u<=n;++u)cin>>st[u][0];
33 L=log(n)/log(2)+1;
34 dfs(1);
35 for(int i=1;i<=m;++i){
36     int x,k;
37     cin>>x>>k;
38     k=min(k,d[x]-1); ←————
39     cout<<query(x,k)<<endl;
40 }
```


$p[u][i]$ 代表 u 的第 2^i 个祖先

$st[u][i]$ 代表从 u 开始往上爬
共 2^i 个节点的点权最值

```
3  const int N=200009;
4  vector<int> son[N];
5  int L,n,m,d[N],p[N][30],st[N][30];
6  void dfs(int u){
7      d[u]=d[p[u][0]]+1;
8      for(int i=1;i<=L;++i){
9          p[u][i]=p[p[u][i-1]][i-1];
10         st[u][i]=max(st[u][i-1],st[p[u][i-1]][i-1]);
11     }
12     for(int i=0;i<son[u].size();++i)
13         dfs(son[u][i]);
14 }
```

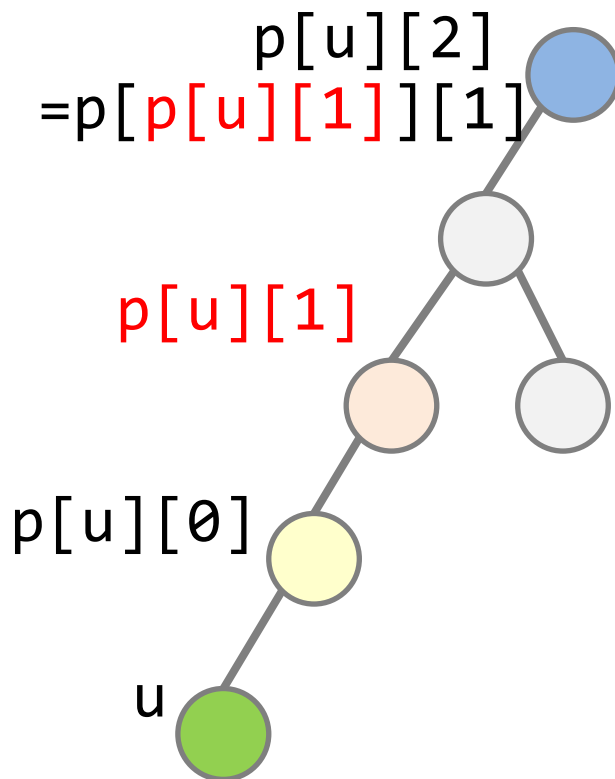
$p[u][i]$ 代表 u 的第 2^i 个祖先

$st[u][i]$ 代表从 u 开始往上爬
共 2^i 个节点的点权最值

```
15 int query(int u, int k){
16     int ans = st[u][0];
17     for(int i = 0; i <= L; ++i)
18         if((1 << i) & k){
19             ans = max(ans, );
20             u = p[u][i];
21         }
22     return ans;
23 }
```

易错点

$st[u][i]$ 不包含 $p[u][i]$



太戈编程

1825

1826

857

拓展题

1011,662