

# ST表与树上倍增LCA 板子

板子

## ST表

```
1.  #include <iostream>
2.  #include <cstdio>
3.  #include <algorithm>
4.  #include <cctype>
5.  #define MAXN 100003
6.  using namespace std;
7.  int st[MAXN][22], log[MAXN];
8.  int n, m;
9.  inline void read(int& i) {
10.     i=0; char c=getchar();
11.     while(!isdigit(c)) c=getchar();
12.     while(isdigit(c)) i=i*10+c-'0', c=getchar();
13. }
14. inline query(int L, int R) {
15.     int k=log[R-L+1];
16.     return max(st[L][k], st[R-(1<<k)+1][k]);
17. }
18. int main() {
19.     read(n); read(m);
20.     log[0]=-1;
21.     for(int i=1; i<=n; ++i) read(st[i][0]);
22.     for(int i=1; i<=n; ++i) log[i]=log[i>>1]+1;
23.     for(int j=1; j<=20; ++j) {
24.         for(int i=1; i+(1<<j)-1<=n; ++i) {
25.             st[i][j]=max(st[i][j-1], st[i+(1<<(j-1))][j-1]);
26.         }
27.     }
28.     int L, R;
29.     while(m--){
30.         read(L), read(R);
31.         printf("%d\n", query(L, R));
32.     }
33.     return 0;
34. }
```

## 倍增LCA

```
1.  #include<bits/stdc++.h>
2.  using namespace std;
3.  const int maxn = 500010,max_dep=25;
4.  vector<int>G[maxn];
5.  int n,m;
6.  int parent[maxn][max_dep];
7.  int dep[maxn];
8.  int root;
9.  inline void addEdges(int u,int v){
10.     G[u].push_back(v);
11.     G[v].push_back(u);
12. }
13. inline void dfs(int rt,int v){
14.     dep[v]=dep[rt]+1;
15.     parent[v][0]=rt;
16.     for(int i=1;i<=19;++i)
17.         parent[v][i]=parent[parent[v][i-1]][i-1];
18.     for(int i=0;i<G[v].size();++i)
19.         if(G[v][i]!=rt)
20.             dfs(v,G[v][i]);
21. }
22. inline int LCA(int u,int v){
23.     if(dep[u]>dep[v])
24.         swap(u,v);
25.     for(int i=0;i<=19;++i)
26.         if((dep[v]-dep[u])>>i&1)
27.             v=parent[v][i];
28.     if(u==v)
29.         return u;
30.     for(int i=19;i>=0;--i)
31.         if(parent[u][i]!=parent[v][i]){
32.             v=parent[v][i];
33.             u=parent[u][i];
34.         }
35.     return parent[u][0];
36. }
37. int main(){
38.     read(n);
39.     read(m);
40.     read(root);
41.     int u,v;
42.     for(int i=1;i<n;++i){
```

```
43.         read(u);
44.         read(v);
45.         addEdges(u,v);
46.     }
47.     dfs(0,root);
48.     for(int i=1;i<=m;++i){
49.         int a,b;
50.         read(a);
51.         read(b);
52.         printf("%d\n",LCA(a,b));
53.     }
54.     return 0;
55. }
```