

最近公共祖先(LCA)

```
#include <bits/stdc++.h>
using namespace std;

struct node{
    int to,next;
}e[500010<<1];

int head[500010],depth[500010],fa[500010][22],lg[500010],tot;
inline void add(int x,int y){
    tot++;
    e[tot].to=y;
    e[tot].next=head[x];
    head[x]=tot;
}

inline void dfs(int now,int fath){
    fa[now][0]=fath;
    depth[now]=depth[fath]+1;
    for(int i=1;i<=lg[depth[now]];i++){
        fa[now][i]=fa[fa[now][i-1]][i-1];
    }
    for(int i=head[now];i;i=e[i].next){
        if(e[i].to!=fath) dfs(e[i].to,now);
    }
}

inline int LCA(int x,int y){
    if(depth[x]<depth[y]) swap(x,y);
    while (depth[x]>depth[y]){
        x=fa[x][lg[depth[x]]-depth[y]-1];
    }
    if(x==y) return x;
    for(int k=lg[depth[x]]-1;k>=0;k--){
        if(fa[x][k]!=fa[y][k]){
            x=fa[x][k],y=fa[y][k];
        }
    }
    return fa[x][0];
}

int main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    int n,m,s,x,y;
    cin >> n >> m >> s;
    for(int i=1;i<=n-1;i++){
        cin >> x >> y;
        add(x,y);add(y,x);
    }
    for(int i=1;i<=n;i++){
        lg[i]=lg[i-1]+(1<<lg[i-1]==i); //常数优化
    }
}
```

```
dfs(s,0);  
for(int i=1;i<=m;i++){  
    cin >> x >> y;  
    cout << LCA(x,y) << endl;  
}  
return 0;  
}
```