**洛谷P3379 【模板】最近公共祖先（LCA）**

**题目描述**

如题，给定一棵有根多叉树，请求出指定两个点直接最近的公共祖先。

**输入输出格式**

**输入格式：**

第一行包含三个正整数N、M、S，分别表示树的结点个数、询问的个数和树根结点的序号。

接下来N-1行每行包含两个正整数x、y，表示x结点和y结点之间有一条直接连接的边（数据保证可以构成树）。

接下来M行每行包含两个正整数a、b，表示询问a结点和b结点的最近公共祖先。

**输出格式：**

输出包含M行，每行包含一个正整数，依次为每一个询问的结果。

**输入输出样例**

**输入样例#1：** 复制

5 5 4

3 1

2 4

5 1

1 4

2 4

3 2

3 5

1 2

4 5

**输出样例#1：** 复制

4

4

1

4

4

**说明**

时空限制：1000ms,128M

数据规模：

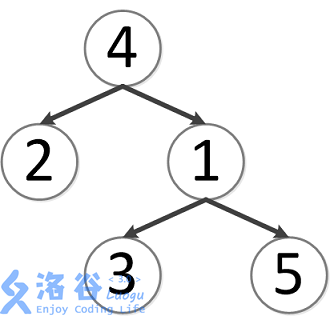
对于30%的数据：N<=10，M<=10

对于70%的数据：N<=10000，M<=10000

对于100%的数据：N<=500000，M<=500000

样例说明：

该树结构如下：



第一次询问：2、4的最近公共祖先，故为4。

第二次询问：3、2的最近公共祖先，故为4。

第三次询问：3、5的最近公共祖先，故为1。

第四次询问：1、2的最近公共祖先，故为4。

第五次询问：4、5的最近公共祖先，故为4。

故输出依次为4、4、1、4、4。

#include<cstdio>

#include<iostream>

#include<cstring>

using namespace std;

const int maxn=500000+2;

int n,m,s;

int k=0;

int head[maxn],d[maxn],p[maxn][21];

//head数组就是链接表标配了吧？d存的是深度（deep）,p[i][j]存的[i]向上走2的j次方那么长的路径

struct node{

int v,next;

}e[maxn\*2];//存树

void add(int u,int v){

e[k].v=v;

e[k].next=head[u];

head[u]=k++;

} //加边函数

void dfs(int u,int fa){

d[u]=d[fa]+1;

p[u][0]=fa;

for(int i=1;i<16;i++)

p[u][i]=p[p[u][i-1]][i-1];

for(int i=head[u];i!=-1;i=e[i].next){

int v=e[i].v;

if(v!=fa)

dfs(v,u);

}

} //首先进行的预处理，将所有点的deep和p的初始值dfs出来

int lca(int a,int b){ //非常标准的lca查找

if(d[a]>d[b])

swap(a,b); //保证a是在b结点上方，即a的深度小于b的深度

for(int i=16;i>=0;i--)

if(d[a]<=d[b]-(1<<i))

b=p[b][i]; //先把b移到和a同一个深度

if(a==b)

return a; //特判，如果b上来和就和a一样了，那就可以直接返回答案了

for(int i=16;i>=0;i--){

if(p[a][i]==p[b][i])

continue;

else

a=p[a][i],b=p[b][i]; //A和B一起上移

}

return p[a][0]; // 找出最后a值的数字

}

int main(){

memset(head,-1,sizeof(head));

memset(p,-1,sizeof(p));

int a,b;

scanf("%d%d%d",&n,&m,&s);

for(int i=1;i<n;i++){

scanf("%d%d",&a,&b);

add(a,b);

add(b,a); //无向图，要加两次

}

dfs(s,0);

for(int i=1;i<=m;i++){

scanf("%d%d",&a,&b);

printf("%d\n",lca(a,b));

}

return 0;

}