MMSet2(LCA&树的直径)

树剖LCA。

思路：树的直径，结论：一棵树的任意点集S中距离最远的两个点(类似树的直径)必有一个点是深度最大的点，且树上所有点的到该节点距离最大值的最小值为：该点集的直径除以2向上取整。

因为：所有点的$f(u)$都是与该点集直径$d(x,y)$到这两个点的距离取mx，所以要找到最小的值，只需要$\frac{mx+1}{2}$，这样肯定是最小的，其他的都会大于距离的一半。

$ep:mx=d(x,y)=5$,肯定取$path(x,y)$路径上中间的点，这样$max(dis(x,u),dis(y,u))$是最小的。

```
x
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=3e5+5,M=1e6+5;
#define mst(a,b) memset(a,b,sizeof a)
#define lx x<<1
#define rx x<<1|1
#define reg register
#define PII pair<int,int>
#define fi first
#define se second
#define pb push_back
#define il inline
template<class T>
inline void read(T &x){
    x=0;int w=1;
    char ch=getchar();
    while(ch<'0'||ch>'9') {if(ch=='-') w=-1;ch=getchar();}
    for(;ch>='0'&&ch<='9';ch=getchar())
        x=(x<<3)+(x<<1)+(ch&15);
    x*=w;
}
struct edge{
    int to,nt;
}e[N<<1];
int n,q,h[N],sz[N],son[N],top[N],cnt,fa[N],dep[N];
```

```
int a[M];
il void add(int u,int v){
    e[++cnt].to=v,e[cnt].nt=h[u],h[u]=cnt;
}
void dfs1(int u,int f){ //求size,dep,重儿子son
    sz[u]=1,fa[u]=f,dep[u]=dep[f]+1;
    for(int i=h[u];i;i=e[i].nt){
         int v=e[i].to;
         if(v==f) continue;
         dfs1(v,u),sz[u]+=sz[v];
         if(sz[v]>sz[son[u]]) son[u]=v;
    }
}
void dfs2(int u,int tp){ //求链顶top
     top[u]=tp;
     if(son[u]) dfs2(son[u],tp);
     for(int i=h[u];i;i=e[i].nt){
            int v=e[i].to;
           if(v!=fa[u]&&v!=son[u]) dfs2(v,v);
     }
}
int lca(int x,int y){ //树剖LCA
     while(top[x]!=top[y]){
         if(dep[top[x]]<dep[top[y]]) swap(x,y);
         x=fa[top[x]];
     }
     return dep[x]<dep[y]?x:y;
}
il int dis(int x,int y){ //两点距离.
    return dep[x]+dep[y]-2*dep[lca(x,y)];
}
int main(){
    read(n);
    for(int i=1,u,v;i<n;i++){
        read(u),read(v);
        add(u,v),add(v,u);
    }
```

```
    dfs1(1,0),dfs2(1,1);
    read(q);
    while(q--){
        int m,u=-1,mx=0;
        read(m);
        for(int i=1;i<=m;i++){
            read(a[i]);
            if(u==-1||dep[a[i]]>dep[u]) u=a[i];
        }
        for(int i=1;i<=m;i++){
            int tmp=dis(u,a[i]);
            if(tmp>mx) mx=tmp;
        }
        printf("%d\n",(mx+1)>>1);
    }
    return 0;
}
```

P3379 【模板】最近公共祖先（LCA）

倍增LCA。

[题目传送门]

思路：模板题思路不多说，唯一需要注意的一点是此题用vector会多耗费时间（因为内存不够会自动申请2倍内存，然后复制元素到新内存耗费时间），用链式快很多。具体见代码。

```
xxxxxxxxxx
#include<bits/stdc++.h>
using namespace std;
const int N=1e6+5;
const int M=log2(N);
int fa[N][M],dep[N],mx,cnt=1,h[N];
struct edge{
    int to,nt;
}e[N];
void add(int u,int v){
    e[cnt].to=v;
    e[cnt].nt=h[u];
```

```
    h[u]=cnt++;
}
void dfs(int u){
    for(int i=1;i<=mx;i++)
        if(fa[u][i-1]) //如果u不存在第2^(i-1)父亲,则肯定不存在fa[u][i]
            fa[u][i]=fa[fa[u][i-1]][i-1];
        else break;
    for(int i=h[u];i;i=e[i].nt){
        int v=e[i].to;
        if(v!=fa[u][0]){
            fa[v][0]=u;
            dep[v]=dep[u]+1;
            dfs(v);
        }
    }
}
int lca(int u,int v){
    if(dep[u]<dep[v]) swap(u,v);//默认u为较大深度的结点
    int delta=dep[u]-dep[v];//深度差值.
    for(int i=0;i<=mx;i++) //将u提到与v同深度
        if((1<<i)&delta)
            u=fa[u][i];
    if(u==v) return u;//入股此时相等说明u,v的祖先就是u(v)
    for(int i=mx;i>=0;i--)//这里要从大往小提,来达到越提越精确的效果。
        if(fa[u][i]!=fa[v][i]) //如果父亲不相等就继续一起往上题
        {
            u=fa[u][i];
            v=fa[v][i];
        }
    return fa[u][0];//这是u,v的父亲就是他们的LCA
}
int main(){
    int n,m,s,a,b;
    scanf("%d%d%d",&n,&m,&s);
    mx=log2(n);//最大移动
    for(int i=1;i<n;i++){
            scanf("%d%d",&a,&b);
```

```
            add(a,b);
            add(b,a);
    }
    dfs(s);//从根结点搜
    while(m--){
        scanf("%d%d",&a,&b);
        printf("%d\n",lca(a,b));
    }
    return 0;
}
```