

运输计划

一. 考察内容:

树 树链剖分 二分答案

二. 题目分析:

[题目大意]

将一棵树的一条边的边权设为0, 使得所有询问的两点之间路径最长的最短。

[写题思路]

考虑这道题让求出最长路最短, 我们会想到二分答案的做法。

首先预处理出所有询问的路径长度, 我们二分出一个答案, 将比该答案长的路径所需要经过的边全部打上标记, 最终可以求出这些询问的公共边 (上面的标记数为比答案长的路径数)。如果有一条公共边的长度, 大于等于 (这些路径中最长的长度-二分答案), 那么该答案就是可行的 (由于减掉这条路径之后, 最长的路径也不会超过答案, 所以答案可行), 否则就不可行。

三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATE
/*****
*创建时间: 2018 08 26
*文件类型: 源代码文件
*题目来源: 洛谷
*当前状态: 已通过
*备忘录: 图论 二分答案 最短路
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
// #include <sys/wait.h>
// #include <sys/types.h>
// #include <unistd.h>
using namespace std;
namespace IO
{
    #define SIZE 10000
    char s[SIZE+1];
    int T=0, end=0;
    int in()
    {
        T=0;
        return end=fread(s, 1, SIZE, stdin);
    }
    int read(int &a)
    {
        int i=a=0, flag=1;
        while(1)
        {
            if(T==end&&!in())return i;
            if(s[T]=='\r' || s[T]=='\n' || s[T]==' ')
            {
                if(i) return i;
            }
            else if(s[T]=='-')flag=-1;
            else

```

```

        {
            ++i;
            a=flag*(flag*10*a+s[T]-'0');
        }
        ++T;
    }
}

#define MAXN 3000100
int n,m,head[MAXN],_edge,
son[MAXN],bro[MAXN],fa[MAXN],fav[MAXN],
num[MAXN],top[MAXN],deep[MAXN],d[MAXN],h_son[MAXN],dfsx[MAXN],_dfsx,
diff[MAXN];
struct EDGE
{
    int a,b,v,next;
    EDGE(int a=0,int b=0,int v=0,int next=0):a(a),b(b),v(v),next(next){}
}edge[MAXN];
struct A
{
    int a,b,lca,d;
}a[MAXN];
/* Variable explain:

*/
void adde(int a,int b,int v)
{
    edge[++_edge]=EDGE(a,b,v,head[a]);
    head[a]=_edge;
}
void addt(int a,int b,int v)
{
    bro[b]=son[a];
    son[a]=b;
    fa[b]=a;
    fav[b]=v;
}
void read()
{
    using I0::read;
    int ls1,ls2,ls3;
    // freopen("transport.in","r",stdin);
    // freopen("transport.out","w",stdout);
    read(n);
    read(m);
    for(int i=1;i<n;+
+i) read(ls1),read(ls2),read(ls3),adde(ls1,ls2,ls3),adde(ls2,ls1,ls3);
    for(int i=1;i<=m;++i) read(a[i].a),read(a[i].b);
    return;
}
void dfs1(int a,int c,int nowd)
{
    deep[a]=c;
    d[a]=nowd;
    dfsx[a]=++_dfsx;
    for(int i=head[a];i;i=edge[i].next)
    {
        int b=edge[i].b,v=edge[i].v;
        if(dfsx[b]) continue;
        addt(a,b,v);
        dfs1(b,c+1,nowd+v);
        num[a]+=num[b];
        if(num[b]>num[h_son[a]]) h_son[a]=b;
    }
    ++num[a];
}
void dfs2(int a,int t)
{
    dfsx[a]=++_dfsx;
    top[a]=t;
    if(h_son[a]) dfs2(h_son[a],t);
    for(int i=son[a];i;i=bro[i])

```

```

        if(i!=h_son[a])dfs2(i,i);
    }
    inline int lca(int a,int b)
    {
        while(1)
        {
            if(top[a]==top[b])return deep[a]<deep[b]?a:b;
            if(deep[top[a]]>deep[top[b]])swap(a,b);
            b=fa[top[b]];
        }
    }
    void dfs3(int a)
    {
        for(int i=son[a];i;i=bro[i])
        {
            dfs3(i);
            diff[a]+=diff[i];
        }
    }
    inline bool check(int x)
    {
        int counts=0,max_need=0;
        for(register int i=1;i<=n;++i)diff[i]=0;
        for(register int i=1;i<=m;++i)
        {
            if(a[i].d>x)
            {
                ++diff[a[i].a],++diff[a[i].b],diff[a[i].lca]-=2;
                ++counts;
                max_need=max(max_need,a[i].d-x);
            }
        }
        dfs3(1);
        for(register int i=1;i<=n;++i)if(diff[i]==counts&&fav[i]>=max_need)return true;
        return false;
    }
    void solve()
    {
        int l=0,r=3e8,ans=0;
        while(l<=r)
        {
            int m=(l+r)>>1;
            if(check(m))r=m-1,ans=m;
            else l=m+1;
        }
        printf("%d\n",ans);
    }
    int main()
    {
        read();
        dfs1(1,0,0);
        _dfsx=0;
        dfs2(1,1);
        // for(int i=1;i<=n;++i)printf("%d ",dfsx[i]);puts("");
        // for(int i=1;i<=n;++i)printf("%d ",top[i]);puts("");
        for(register int i=1;i<=m;++i)a[i].lca=lca(a[i].a,a[i].b),a[i].d=d[a[i].a]
+d[a[i].b]-(d[a[i].lca]<<1);
        // for(int i=1;i<=m;++i)printf("%d ",a[i].d);
        solve();
        return 0;
    }
}

```

[<题目跳转>](#) [<查看代码>](#)