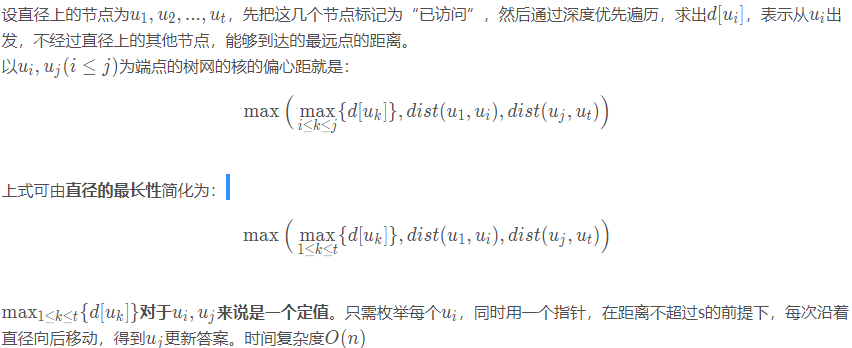


NOIP2007提高组 树网的核

【题目大意】

求无根树的直径上一段不超过S长的链，使得偏心距最小。具体概念见[原题](http://www.lydsy.com/JudgeOnline/problem.php?id=1999" \t "_blank)。



#include<cstdio>

#include<cstring>

#include<algorithm>

using namespace std;

const int N=5e5+10;

struct Edge{

int v,nx,w;

}e[N<<1];

int n,s,hd[N],tot,fa[N],vis[N],dis[N];

void addedge(int u,int v,int w)

{

e[tot].v=v;

e[tot].w=w;

e[tot].nx=hd[u];

hd[u]=tot++;

}

void dfs(int u,int f)

{

fa[u]=f;

for(int i=hd[u];~i;i=e[i].nx)

{

int v=e[i].v;

if(vis[v]||v==f)continue;

dis[v]=dis[u]+e[i].w;

dfs(v,u);

}

}

int main()

{

//freopen("in.txt","r",stdin);

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

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

int u,v,w;

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

{

scanf("%d%d%d",&u,&v,&w);

addedge(u,v,w);addedge(v,u,w);

}

int l=1,r=1;

dfs(1,0);

for(int i=1;i<=n;i++)if(dis[i]>dis[r])r=i;

l=r;

dis[r]=0;

dfs(r,0);

for(int i=1;i<=n;i++)if(dis[i]>dis[l])l=i;

int i=l,j=l;

int ans=0x7fffffff;

for(;i;i=fa[i])

{

while(fa[j]&&dis[i]-dis[fa[j]]<=s)j=fa[j];

ans=min(ans,max(dis[j],dis[l]-dis[i]));

}

for(i=l;i;i=fa[i])vis[i]=1;

for(i=l;i;i=fa[i])dis[i]=0,dfs(i,fa[i]);

for(i=1;i<=n;i++)ans=max(ans,dis[i]);

printf("%d\n",ans);

return 0;

}

## 单调队列

//O(n)

#include<cstdio>

#include<cstring>

#include<algorithm>

#include<queue>

using namespace std;

typedef long long ll;

typedef unsigned long long ull;

typedef pair<int,int> pii;

typedef pair<ll,ll> pll;

void sort(int &a,int &b){

if(a>b)

swap(a,b);

}

int rd(){

int s=0,c;

while((c=getchar())<'0'||c>'9');

do {

s=s\*10+c-'0';

}

while((c=getchar())>='0'&&c<='9');

return s;

}

int upmin(int &a,int b){

if(b<a) {

a=b;

return 1;

}

return 0;

}

int upmax(int &a,int b){

if(b>a) {

a=b;

return 1;

}

return 0;

}

struct graph{

int v[1000010];

int w[1000010];

int t[1000010];

int h[500010];

int n;

graph(){

memset(h,0,sizeof h);

n=0;

}

void add(int x,int y,int z){

n++;

v[n]=y;

w[n]=z;

t[n]=h[x];

h[x]=n;

}

};

graph g;

int from[500010];

int b[500010];

queue<int> q;

int d[500010];

int w[500010];

void bfs(int x){

memset(d,-1,sizeof d);

d[x]=0;

q.push(x);

int i;

from[x]=0;

w[x]=0;

while(!q.empty()){

x=q.front();

q.pop();

for(i=g.h[x];i;i=g.t[i])

if(d[g.v[i]]==-1){

d[g.v[i]]=d[x]+g.w[i];

w[g.v[i]]=g.w[i];

from[g.v[i]]=x;

q.push(g.v[i]);

}

}

}

void dfs(int x,int fa,int dep,int &s){

s=max(s,dep);

int i;

for(i=g.h[x];i;i=g.t[i])

if(!b[g.v[i]]&&g.v[i]!=fa)

dfs(g.v[i],x,dep+g.w[i],s);

}

int a[500010];

int c[500010];

int f[500010];

int fl[500010];

int fr[500010];

pii q2[500010];

int head,tail;

void add(int x){

while(tail>=head&&q2[tail].second<=f[x])

tail--;

q2[++tail]=pii(x,f[x]);

}

void del(int x){

if(q2[head].first==x)

head++;

}

int main(){

freopen("bzoj1999.in","r",stdin);

freopen("bzoj1999.out","w",stdout);

int n,m;

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

int i,x,y,z;

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

scanf("%d%d%d",&x,&y,&z);

g.add(x,y,z);

g.add(y,x,z);

}

bfs(1);

x=1;

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

if(d[i]>d[x])

x=i;

bfs(x);

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

if(d[i]>d[x])

x=i;

int t=0;

do {

a[++t]=x;

c[t]=w[x];

x=from[x];

} while(x);

for(i=1;i<=t;i++)

b[a[i]]=1;

int j;

for(i=1;i<=t;i++)

for(j=g.h[a[i]];j;j=g.t[j])

if(!b[g.v[j]])

dfs(g.v[j],0,g.w[j],f[i]);

for(i=1;i<=t;i++)

fl[i]=max(f[i],fl[i-1]+c[i-1]);

for(i=t;i>=1;i--)

fr[i]=max(f[i],fr[i+1]+c[i]);

for(i=1;i<=t;i++)

c[i]+=c[i-1];

head=1;

tail=0;

j=1;

int ans=0x7fffffff;

for(i=1;i<=t;i++) {

add(i);

while(c[i-1]-c[j-1]>m){

del(j);

j++;

}

ans=min(ans,max(max(fl[j],fr[i]),q2[head].second));

}

printf("%d\n",ans);

return 0;

}