目录

1 新的板子

1.1 DLX

```
/*
    嵌套变量重复 特殊样例 0 1 2 n 数组越界 开long long
    清空 建图别用vector
*/
#include<bits/stdc++.h>
using namespace std;
#define endl ”\n”
#define pp(x) array<int,x>
using ull=unsigned long long;
using ll=long long;
using pii=pair<int,int>;
const int dx[]={0,0,1,-1,1,-1,1,-1};
const int dy[]={1,-1,0,0,1,-1,-1,1};
const int mod=998244353;
const int inf=0x3f3f3f3f;
const int INF=1e9+7;
const int maxn=1e6+100;
const int MaxM=450010;
const int MaxP=510;
struct DLX{
    int U[MaxM],D[MaxM],L[MaxM],R[MaxM];
    int Row[MaxM],Col[MaxM];
    int First[MaxP];
    int Size[MaxM];
    int tot=0;
    int ans;
    void build(int n,int num){
        for(int i=0;i<=num;i++){
            L[i]=i-1;R[i]=i+1;
            U[i]=i;D[i]=i;
            Size[i]=0;
        }
        L[0]=num;
        R[num]=0;
        tot=num;
        for(int i=1;i<=n;i++)
            First[i]=-1;
    }
    void insert(int r,int c){
        Row[++tot]=r; Col[tot]=c;
        D[tot]=D[c]; U[D[c]]=tot;
        U[tot]=c; D[c]=tot;
        if(First[r]<0){
            L[tot]=tot;
            R[tot]=tot;
            First[r]=tot;
        }
        else{
            R[tot]=R[First[r]];
            L[R[First[r]]]=tot;
            L[tot]=First[r];
            R[First[r]]=tot;
        }
        Size[c]++;
    }
    void remove(int c){
        L[R[c]]=L[c];
        R[L[c]]=R[c];
        for(int i=D[c];i!=c;i=D[i]){
            for(int j=R[i];j!=i;j=R[j]){
                U[D[j]]=U[j];
                D[U[j]]=D[j];
                Size[Col[j]]--;
            }
        }
    }
    void resume(int c){
        for(int i=U[c];i!=c;i=U[i]){
            for(int j=L[i];j!=i;j=L[j]){
                D[U[j]]=j;
                U[D[j]]=j;
                Size[Col[j]]++;
            }
        }
        R[L[c]]=c;
        L[R[c]]=c;
    }
    void dance(int depth){
        if(depth>=ans)
            return ;
        if(R[0]==0){
            ans=depth;
            return ;
        }
        int c=R[0];
        for(int i=R[0];i!=0;i=R[i])
            if(Size[i]<Size[c])
                c=i;
        remove(c);
        for(int i=D[c];i!=c;i=D[i]){
            for(int j=R[i];j!=i;j=R[j])
                remove(Col[j]);
            dance(depth+1);
            for(int j=L[i];j!=i;j=L[j])
                resume(Col[j]);
        }
        resume(c);
    }
}dlx;
int n,m,p;
void solve()
{
    cin>>n>>m>>p;
    dlx.build(p,n*m);
    for(int i=1;i<=p;i++)
    {
        int x,y,_x,_y;
        cin>>x>>y>>_x>>_y;
        for(int I=x+1;I<=_x;I++)
            for(int J=y+1;J<=_y;J++)
                dlx.insert(i,(I-1)*m+J);
    }
    dlx.ans=inf;
    dlx.dance(0);
    if(dlx.ans==inf)cout<<-1<<endl;
    else cout<<dlx.ans<<endl;
}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(nullptr);cout.tie(nullptr);
    int __;cin>>__;
    while(__--)
        solve();
    return 0;
}
```

1.2 AC 自动机

```
#include<bits/stdc++.h>
using namespace std;
#define endl ”\n”
#define int long long
#define pp(x) array<int,x>
using ull=unsigned long long;
using ll=long long;
using pii=pair<int,int>;
using pdd=pair<double,double>;
const int dx[]={0,0,1,-1,1,-1,1,-1};
const int dy[]={1,-1,0,0,1,-1,-1,1};
const int mod=998244353;
const int inf=0x3f3f3f3f;
const int INF=1e9+7;
const int maxn=1e6+100;
struct ACAM{
    int ch[maxn][26],nxt[maxn];
    int val[maxn],cnt[maxn];
    int id[maxn];
    int in[maxn];
    int root=0,indx=0;
    void clear(){
        for(int i=0;i<=indx;i++){
            val[i]=cnt[i]=id[i]=nxt[i]=0;
            for(int j=0;j<26;j++)
                ch[i][j]=0;
        }
        indx=0;
    }
    void insert(const string &str,int x)
    {
        int rt=0;
        for(int i=0;i<str.size();i++){
            int tmp=str[i]-’a’;
            if(!ch[rt][tmp])ch[rt][tmp]=++indx;
            rt=ch[rt][tmp];
        }
        val[rt]++;
        id[x]=rt;
    }
    void build()
    {
        queue<int>q;
        for(int i=0;i<26;i++)
            if(ch[0][i])
                q.push(ch[0][i]);
        while(!q.empty()){
            int x=q.front();q.pop();
            for(int i=0;i<26;i++){
                int &rt=ch[x][i];
                if(!rt)
                    rt=ch[nxt[x]][i];
                else{
                    nxt[rt]=ch[nxt[x]][i];
                    in[ch[nxt[x]][i]]++;
                    q.push(rt);
                }
            }
        }
    }
    void query(const string&s,int n)
    {
        int rt=0;
        for(int i=0;i<s.size();i++)
        {
            int tmp=s[i]-’a’;
            rt=ch[rt][tmp];
            cnt[rt]++;
        }
        queue<int>q;
        for(int i=1;i<=indx;i++)
            if(!in[i])q.push(i);
        while(!q.empty())
        {
            int x=q.front();q.pop();
            int y=nxt[x];
            cnt[y]+=cnt[x];
            in[y]--;
            if(!in[y])q.push(y);
        }
        for(int i=1;i<=n;i++)
            cout<<cnt[id[i]]<<endl;
    }
}ac;
int n;
void solve()
{
    cin>>n;
    for(int i=1;i<=n;i++){
        string str;cin>>str;
        ac.insert(str,i);
    }
    string t;
    cin>>t;
    ac.build();
    ac.query(t,n);
}
signed main(){
 // freopen(”data.in”,”r”,stdin);
 // freopen(”data.out”,”w”,stdout);
    ios::sync_with_stdio(false);
    cin.tie(nullptr);cout.tie(nullptr);
    // int __;cin>>__;
    // while(__--)
        solve();
    return 0;
}
```

1.3 HASH

```
/*
双哈希 测试
https://ac.nowcoder.com/acm/contest/64384/D
*/
#pragma GCC optimize(2)
#pragma GCC optimize(3,”Ofast”,”inline”)
#include<bits/stdc++.h>
using namespace std;
#define endl ”\n”
using ull=unsigned long long;
using ll=long long;
using pii=pair<int,int>;
using PLL=pair<ull,ull>;
const int maxn=1e6+200;
char s[maxn];
int n,m,k;
namespace EA{
    class Hash{
    public:
        ull Prime_Pool[3] = {233317ull,1998585857ul,23333333333ul};
        ull Seed_Pool[4]={911,146527,19260817,91815541};
        ull Mod_Pool[5]={29123,998244353,1000000009,4294967291ull,212370440130137957ll};
        ull sum[maxn],bas[maxn];
        int perm[maxn];
        int sigma;
        ull Seed,Mod;
        int N;
        Hash(int x,int y):Seed(Seed_Pool[x]),Mod(Mod_Pool[y]){
            bas[0]=1;
            for (int i = 1; i <= N; i++){
                bas[i] = bas[i - 1] * Seed % Mod;
            }
        }
        void init(char *s){
            N=strlen(s+1);
            for(int i=1;i<=N;i++)
                sum[i]=(sum[i-1]*Seed%Mod+s[i])%Mod;
            bas[0]=1;
            for(int i=1;i<=N;i++)
                bas[i]=bas[i-1]*Seed%Mod;
        }
        void indexInit(char *s){
            N=strlen(s+1);
            sigma=N+100;
            iota(perm+1,perm+1+sigma,1);
            random_shuffle(perm+1,perm+1+sigma);
            // for(int i=1;i<=N;i++)
            //     cout<<perm[i]<<” ”;
            // cout<<endl;
            for(int i=1;i<=N;i++)
                sum[i]=(sum[i-1]*Seed%Mod+perm[s[i]])%Mod;
            bas[0]=1;
            for(int i=1;i<=N;i++)
                bas[i]=bas[i-1]*Seed%Mod;
        }
        ull get(int l,int r){
            return (sum[r]-sum[l-1]*bas[r-l+1]%Mod+Mod)%Mod;
        }
    }h[2]{Hash(2,2),Hash(3,3)};
    void solve()
    {
        cin>>n>>m>>k>>s+1;
        h[0].init(s);
        h[1].init(s);
        // h[0].indexInit(s);
        // h[1].indexInit(s);
        map<PLL,int>cnt;
        map<PLL,int>vis;
        for(int i=m;i<=n;i++)
        {
            ull now[2];
            now[0]=h[0].get(i-m+1,i);
            now[1]=h[1].get(i-m+1,i);
            PLL fl={now[0],now[1]};
            if(vis.count(fl)&&vis[fl]<i-m+1)
            {
                cnt[fl]++;
                vis[fl]=i;
            }
            else if(!vis.count(fl))
            {
                cnt[fl]=1;
                vis[fl]=i;
            }
        }
        int ans=0;
        for(auto [x,y]:cnt)
            if(y==k)ans++;
        cout<<ans<<endl;
    }
};
namespace EB{
    class StringHash{
    public:
        ull Mod_Pool[5]={29123,998244353,1000000009,4294967291ull,212370440130137957ll};
        ull Prime_Pool[3] = {233317ull,1998585857ul,23333333333ul};
        ull seed[4]={911,146527,19260817,91815541};
        ull mod=212370440130137957ll;
        ull prime=233317;
        ull base=131;
        vector<ull>h;
        vector<ull>bas;
        StringHash():h(1),bas(1,1),base(131){}
        StringHash(int x,int y):h(1),bas(1,1){base=seed[x];mod=Mod_Pool[y];}
        void push_back(char ch){
            h.push_back((h.back()*base+ch)%mod + prime);
            bas.push_back(bas.back()*base%mod);
        }
        // ull get(int l,int r){
        //  return (h[r] + __int128(h[l])*(mod-bas[r-l]))%mod;
        // }
        ull get(int l,int r){
            return (h[r]-h[l-1]*bas[r-l+1]%mod+mod)%mod;
        }
    };
    void A()
    {
        string str;
        StringHash hs,rhs;
        int N=int(str.size());
        for(int i=0;i<N;i++)
            hs.push_back(str[i]);
        for(int i=N-1;i>=0;i--)
            rhs.push_back(str[i]);
    }
    void solve()
    {
        cin>>n>>m>>k>>s+1;
        StringHash h[2]{StringHash(1,2),StringHash(2,3)};
        for(int i=1;i<=n;i++)
        {
            h[0].push_back(s[i]);
            h[1].push_back(s[i]);
        }
        map<PLL,int>cnt;
        map<PLL,int>vis;
        for(int i=m;i<=n;i++)
        {
            ull now[2];
            now[0]=h[0].get(i-m+1,i);
            now[1]=h[1].get(i-m+1,i);
            PLL fl={now[0],now[1]};
            if(vis.count(fl)&&vis[fl]<i-m+1)
            {
                cnt[fl]++;
                vis[fl]=i;
            }
            else if(!vis.count(fl))
            {
                cnt[fl]=1;
                vis[fl]=i;
            }
        }
        int ans=0;
        for(auto [x,y]:cnt)
            if(y==k)ans++;
        cout<<ans<<endl;
    }
};
namespace shuang{
    class DoubleHash{
    public:
        const ull b1=137,b2=149,i1=1'603'801'661,i2=1'024'053'074;
        const ull p1=2'034'452'107,p2=2'013'074'419;
        ull m1[maxn],m2[maxn],r1,r2;
        PLL h[maxn];
        int N;
        void init(char*s){
            h[0]={0,0};
            int n=strlen(s+1);
            m1[0]=m2[0]=1;
            N=n+10;
            for(int i=1;i<=N;i++)
            {
                m1[i]=m1[i-1]*b1%p1;
                m2[i]=m2[i-1]*b2%p2;
            }
            h[0]={0,0};
            for(int i=1;i<=n;i++)
            {
                r1=(r1+s[i]*m1[i])%p1;
                r2=(r2+s[i]*m2[i])%p2;
                h[i]={r1,r2};
            }
        }
        PLL getv(int l,int r){
            return {(p1+h[r].first-h[l].first)*m1[N-l]%p1,(p2+h[r].second-h[l].second)*m2[N-l]%p2};
        }
        ull get(int l,int r){
            PLL a=getv(l,r);
            return (ull)a.first<<32|a.second;
        }
    }h;
};
signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(nullptr);cout.tie(nullptr);
    EA::solve();
    return 0;
}
```

1.4 SAM 后缀自动机

```
#pragma comment(linker, ”/STACK:1024000000,1024000000”)
#include<bits/stdc++.h>
using namespace std;
#define endl ”\n”
#define int long long
#define pp(x) array<int,x>
using ull=unsigned long long;
using ll=long long;
using pii=pair<int,int>;
using pdd=pair<double,double>;
const int dx[]={0,0,1,-1,1,-1,1,-1};
const int dy[]={1,-1,0,0,1,-1,-1,1};
const int mod=998244353;
const int inf=0x3f3f3f3f;
const int INF=1e9+7;
const int maxn=1e6+100;
class SAM{
public:
    vector<int>v[maxn];
    int ch[maxn][26],siz[maxn],len[maxn],fa[maxn];
    bitset<maxn>vis;
    int f[maxn];
    int tot=1,np=1;
    ll ans;
    void clear(){
        for(int i=0;i<=tot;i++){
            for(int j=0;j<26;j++)
                ch[i][j]=0;siz[i]=len[i]=fa[i]=vis[i]=0;
        }tot=np=1;
    }
    void insert(const int&c){
        int p=np; np=++tot;
        len[np]=len[p]+1; siz[np]=1;
        for(;p&&!ch[p][c];p=fa[p])ch[p][c]=np;
        if(p==0)fa[np]=1;
        else{
            int q=ch[p][c];
            if(len[q]==len[p]+1)fa[np]=q;
            else{
                int nq=++tot;
                len[nq]=len[p]+1;
                fa[nq]=fa[q]; fa[q]=nq; fa[np]=nq;
                for(;p&&ch[p][c]==q;p=fa[p])ch[p][c]=nq;
                memcpy(ch[nq],ch[q],sizeof(ch[q]));
            }
        }
    }
    void insert(const char*s){for(int i=0;s[i]!='\0';i++)insert(s[i]-'a');}
    void insert(const string&s){for(int i=0;i<(int)s.size();i++)insert(s[i]-'a');}
    void build_tree(){
        for(int i=2;i<=tot;i++)
            v[fa[i]].push_back(i);
    }
    /*  op=0本质不同子串 op=1位置不同子串  */
    void build(int op=0){
        build_tree();
        if(!op)
        {
            for(int i=1;i<=tot;i++)
                f[i]=siz[i]=1;
        }
        else get_siz(1);
        f[1]=siz[1]=0;
        get_f(1);
    }
    void get_kth(int k,int rt=1)
    {
        if(k>f[rt]){
            cout<<-1<<endl;
            return ;
        }
        if(k<=siz[rt]){cout<<”\n”;return ;}
        k-=siz[rt];
        if(k==0){return;}
        for(int i=0;i<26;i++){
            if(ch[rt][i]){
                if(f[ch[rt][i]]>=k){
                    cout<<(char)(i+’a’);
                    get_kth(k,ch[rt][i]);
                    return ;
                }
                else
                    k-=f[ch[rt][i]];
            }
        }
        cout<<-1<<endl;
    }
    void match(const string&s)
    {
        // queue<int>q;
        // for(int i=1;i<=tot;i++){
        //  f[i]=0,in[i]=main_in[i];
        //  if(!in[i])q.push(i);
        // }
        // int rt=1;
        // int length=0;
        // for(int i=0,tmp;i<(int)s.size();i++)
        // {
        //  tmp=s[i]-’a’;
        //  while(rt&&!ch[rt][tmp])
        //      rt=fa[rt],length=len[rt];
        //  if(!rt)rt=1;
        //  else{
        //      rt=ch[rt][tmp];length++;
        //  }
        //  f[rt]=max(f[rt],length);
        // }
        // while(!q.empty()){
        //  int x=q.front();q.pop();
        //  mi[x]=min(mi[x],f[x]);
        //  int y=fa[x];
        //  if(f[x]>=len[y])
        //      f[y]=len[y];
        //  in[y]--;
        //  if(!in[y])q.push(y);
        // }
    }
    void debug(){
        for(int i=0;i<=tot;i++)
            for(int j=0;j<26;j++)
                if(ch[i][j])
                    cout<<”ins␣”<<i<<”␣--”<<(char)(j+’a’)
                       <<”-->␣”<<ch[i][j]<<endl;
        for(int i=1;i<=tot;i++)
            cout<<”fail␣”<<i<<”␣->␣”<<fa[i]<<endl;
        for(int i=1;i<=tot;i++)
            cout<<”len[”<<i<<”]=”<<len[i]<<endl;
    }
    void query();
private:
    void get_siz(int x){
        for(auto y:v[x]){
            get_siz(y);
            siz[x]+=siz[y];
        }
        f[x]=siz[x];
    }
    void get_f(int x){
        if(vis[x])return ;
        vis[x]=1;
        for(int i=0;i<26;i++){
            int y=ch[x][i];
            if(!y)continue;
            get_f(y);
```

```
      f[x]+=f[y];
    }
  }
}sam;
char ch[maxn];
int op,k;
void solve()
{
  cin>>ch;
  sam.insert(ch);
  // sam.debug();
  sam.build(0);
  int q;cin>>q;
  while(q--){
    cin>>k;
    sam.get_kth(k);
  }
}
signed main(){
 // freopen("data.in","r",stdin);
 // freopen("data.out","w",stdout);
  ios::sync_with_stdio(false);
  cin.tie(nullptr);cout.tie(nullptr);
    solve();
  return 0;
}
```

1.5 PAM 回文自动机

```
#include<bits/stdc++.h>
using namespace std;
#define endl ”\n”
#define pp(x) array<int,x>
using ull=unsigned long long;
using ll=long long;
using pii=pair<int,int>;
using pdd=pair<double,double>;
const int dx[]={0,0,1,-1,1,-1,1,-1};
const int dy[]={1,-1,0,0,1,-1,-1,1};
const int mod=998244353;
const int inf=0x3f3f3f3f;
const int INF=1e9+7;
const int maxn=1e6+100;
class PAM{
public:
    int s[maxn],now;
    int ch[maxn][26],fa[maxn],len[maxn],last,tot;
    int num[maxn];
    int pre=0;
    void clear()
    {
        s[0]=len[1]=-1;
        fa[0]=tot=now=1;
        last=len[0]=0;
        memset(ch[0],0,sizeof(ch[0]));
        memset(ch[1],0,sizeof(ch[1]));
    }
    PAM(){clear();}
    int newnode(int length){
        tot++;
        memset(ch[tot],0,sizeof(ch[tot]));
        fa[tot]=num[tot]=0;
        len[tot]=length;
        return tot;
    }
    int get_fail(int rt){
        while(s[now-len[rt]-2]!=s[now-1])rt=fa[rt];
        return rt;
    }
    void insert(int c)
    {
        s[now++]=c;
        int rt=get_fail(last);
        if(!ch[rt][c]){
            int np=newnode(len[rt]+2);
            fa[np]=ch[get_fail(fa[rt])][c];
            ch[rt][c]=np;
            // num[np]=num[fa[np]]+1;
        }
        last=ch[rt][c];
        num[last]++;
    }
    void build(){
        for(int i=tot;i>=2;i--)
            num[fa[i]]+=num[i];
        num[0]=num[1]=0;
    }
    void insert(char*s){while(*s){insert(*s-’a’);s++;}}
    void insert(const string&s){for(auto i:s)insert(i-’a’);}
    void debug(){
        for(int i=0;i<=tot;i++)
            for(int j=0;j<26;j++)
                if(ch[i][j])
                    cout<<”ins␣”<<i<<”␣--”<<(char)(j+’a’)<<”-->␣”<<ch[i][j]<<endl;
        for(int i=0;i<=tot;i++)
            cout<<”len[”<<i<<”]=”<<len[i]<<endl;
        for(int i=0;i<=tot;i++)
            cout<<”fa[”<<i<<”]=”<<fa[i]<<endl;
    }
```

```
}pam;
string s;
void solve()
{
    cin>>s;
    pam.insert(s);
}
signed main(){
 // freopen(”data.in”,”r”,stdin);
 // freopen(”data.out”,”w”,stdout);
    ios::sync_with_stdio(false);
    cin.tie(nullptr);cout.tie(nullptr);
        solve();
    return 0;
}
```

1.6 倍增优化建图

```cpp
int find(int x,int y){
    if(id[x][y])return id[x][y];
    id[x][y]=++Flow::tot;
    if(!y){
        Flow::Add(id[x][y],Flow::t,val[x]);
    }
    else {
        Flow::Add(id[x][y],find(x,y-1),INF);
        Flow::Add(id[x][y],find(f[x][y-1],y-1),INF);
    }
    return id[x][y];
}

void solve()
{
    cin>>n>>m;
    Flow::init();
    for(int i=1;i<n;i++)
    {
        int x,y,w;
        cin>>x>>y>>w;
        v[x].push_back({y,w});
        v[y].push_back({x,w});
    }
    Flow::s=1;Flow::t=2;Flow::tot=2;
    int ans=0;
    dfs(1,0);
    while(m--)
    {
        int x,y,a,b;
        cin>>x>>y>>a>>b;a-=b;
        if(a<=0)continue;
        ans+=a;
        int now=++Flow::tot;
        Flow::Add(Flow::s,now,a);
        int lca=LCA(x,y);
        for(int i=14;i>=0;i--)
            if((dep[x]-dep[lca])>>i&1)
            {
                Flow::Add(now,find(x,i),INF);
                x=f[x][i];
            }
        for(int i=14;i>=0;i--)
            if((dep[y]-dep[lca])>>i&1)
            {
                Flow::Add(now,find(y,i),INF);
                y=f[y][i];
            }
    }
    ans-=Flow::DINIC();
    cout<<ans<<endl;
}
```

1.7 点分治

```
#include<bits/stdc++.h>
using namespace std;
#define endl ”\n”
const int INF=10000005;
const int maxn=1e7+10;
const int N=2e4+10;
struct Edge{int to,next,w;}edge[N];
int head[N],_cnt;
int del[N],siz[N],maxson,root,sum;
int dis[N],d[N],cnt;
int ans[N],q[maxn],judge[maxn],ask[maxn];
int n,m;
void add(int from,int to,int w){
    edge[++_cnt].w=w;
    edge[_cnt].to=to;
    edge[_cnt].next=head[from];
    head[from]=_cnt;
}
void getroot(int x,int fa)
{
    siz[x]=1;
    int sx=0;
    for(int i=head[x];i;i=edge[i].next)
    {
        int y=edge[i].to;
        if(y==fa||del[y])continue;
        getroot(y,x);
        siz[x]+=siz[y];
        sx=max(sx,siz[y]);
    }
    sx=max(sx,sum-siz[x]);
    if(sx<maxson)maxson=sx,root=x;
}
void getdis(int x,int fa)
{
    dis[++cnt]=d[x];
    for(int i=head[x];i;i=edge[i].next)
    {
        int y=edge[i].to;
        if(y==fa||del[y])continue;
        d[y]=d[x]+edge[i].w;
        getdis(y,x);
    }
}
void calc(int x,int w,int op)
{
    cnt=0,d[x]=w;
    getdis(x,0);
    sort(dis+1,dis+1+cnt);
    for(int i=1;i<=m;i++)
    {
        int l=1,r=cnt;
        while(l<r){
            if(dis[l]+dis[r]<=ask[i]){
                if(dis[l]+dis[r]==ask[i])ans[i]+=op;
                ++l;
            }
            else --r;
        }
    }
}
void divide(int x)
{
    calc(x,0,1);
    del[x]=1;
    for(int i=head[x];i;i=edge[i].next)
    {
        int y=edge[i].to;
        if(del[y])continue;
        calc(y,edge[i].w,-1);
        maxson=sum=siz[y];
        getroot(y,0);
        divide(root);
    }
}
char ch[maxn];
void solve()
{
    cin>>n>>m;
    for(int i=1;i<n;i++)
    {
        int x,y,w;
        cin>>x>>y>>w;
        add(x,y,w);
        add(y,x,w);
    }
    for(int i=1;i<=m;i++)
        cin>>ask[i];
    maxson=sum=n;
    getroot(1,0);
    getroot(root,0);
    divide(root);
    for(int i=1;i<=m;i++)
        cout<<(ans[i]?”AYE\n”:”NAY\n”);
}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(nullptr);cout.tie(nullptr);
        solve();
    return 0;
}
```

1.8 仙人掌求环长度

```
#pragma GCC optimize(2)
#pragma GCC optimize(3,”Ofast”,”inline”)
#include<bits/stdc++.h>
using namespace std;
#define endl ”\n”
#define int long long
#define pp(x) array<int,x>
using ull=unsigned long long;
using ll=long long;
using pii=pair<int,int>;
using pdd=pair<double,double>;
const int dx[]={0,0,1,-1,1,-1,1,-1};
const int dy[]={1,-1,0,0,1,-1,-1,1};
const int mod=998244353;
const int inf=0x3f3f3f3f;
const int INF=1e14+7;
const int maxn=4e6+100;
struct Edge{
    int to,w,next;
}edge[maxn];
int head[maxn],cnt;
int dfn[maxn],low[maxn],indx;
stack<pii>s;
int n,m,ans;
void init()
{
    while(!s.empty())s.pop();
    for(int i=1;i<=n;i++)
    {
        dfn[i]=low[i]=head[i]=0;
    }
    cnt=1;
    indx=0;
    ans=INF;
}
```

```
void add(int from,int to,int w)
{
    edge[++cnt].w=w;
    edge[cnt].to=to;
    edge[cnt].next=head[from];
    head[from]=cnt;
}
void tarjan(int x,int fa)
{
    dfn[x]=low[x]=++indx;
    for(int i=head[x];i;i=edge[i].next)
    {
        int y=edge[i].to;
        if(!dfn[y])
        {
            s.push({x,edge[i].w});
            tarjan(y,i);
            low[x]=min(low[x],low[y]);
            if(low[y]>=dfn[x])
            {
                vector<int>g;
                pii v;
                do{
                    v=s.top();s.pop();
                    g.push_back(v.second);
                }while(v.first!=x);

                sort(g.begin(),g.end());

                if(g.size()==1)ans=min(ans,g[0]);
                else if(g.size()==2)ans=min(ans,g[0]+g[1]);
                else ans=min(ans,min(g[0]+g[1],g[2]));
            }
        }
        else if(dfn[x]>dfn[y]&&i!=(fa^1))
        {
            low[x]=min(low[x],dfn[y]);
            s.push({x,edge[i].w});
        }
    }
}
void solve()
{
    cin>>n>>m;
    init();
    for(int i=1;i<=m;i++)
    {
        int x,y,w;
        cin>>x>>y>>w;
        add(x,y,w);
        add(y,x,w);
    }
    tarjan(1,0);
    cout<<ans<<endl;
}
signed main(){
    int size(512<<20); // 512M
    __asm__ ( ”movq␣%0,␣%%rsp\n”::”r”((char*)malloc(size)+size)); // YOUR CODE
    ios::sync_with_stdio(false);
    cin.tie(nullptr);cout.tie(nullptr);
    int __;cin>>__;
    while(__--)
        solve();
    exit(0);
}
```