ACM 模板

Buns_out

2023 年 9 月 15 日

目录

1 DLX 3

2 AC 自动机 7

3 HASH 10

4 SAM 后缀自动机 16

5 PAM 回文自动机 21

6 倍增优化建图 24

7 点分治 26

8 仙人掌求环长度 29

9 DLX 32

1 DLX

```cpp
/*
    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;
```

```
36          for(int i=1;i<=n;i++)
37              First[i]=-1;
38      }
39      void insert(int r,int c){
40          Row[++tot]=r; Col[tot]=c;
41          D[tot]=D[c]; U[D[c]]=tot;
42          U[tot]=c; D[c]=tot;
43          if(First[r]<0){
44              L[tot]=tot;
45              R[tot]=tot;
46              First[r]=tot;
47          }
48          else{
49              R[tot]=R[First[r]];
50              L[R[First[r]]]=tot;
51              L[tot]=First[r];
52              R[First[r]]=tot;
53          }
54          Size[c]++;
55      }
56      void remove(int c){
57          L[R[c]]=L[c];
58          R[L[c]]=R[c];
59          for(int i=D[c];i!=c;i=D[i]){
60              for(int j=R[i];j!=i;j=R[j]){
61                  U[D[j]]=U[j];
62                  D[U[j]]=D[j];
63                  Size[Col[j]]--;
64              }
65          }
66      }
67      void resume(int c){
68          for(int i=U[c];i!=c;i=U[i]){
69              for(int j=L[i];j!=i;j=L[j]){
70                  D[U[j]]=j;
71                  U[D[j]]=j;
72                  Size[Col[j]]++;
```

```
73              }
74          }
75          R[L[c]]=c;
76          L[R[c]]=c;
77      }
78      void dance(int depth){
79          if(depth>=ans)
80              return ;
81          if(R[0]==0){
82              ans=depth;
83              return ;
84          }
85          int c=R[0];
86          for(int i=R[0];i!=0;i=R[i])
87              if(Size[i]<Size[c])
88                  c=i;
89          remove(c);
90          for(int i=D[c];i!=c;i=D[i]){
91              for(int j=R[i];j!=i;j=R[j])
92                  remove(Col[j]);
93              dance(depth+1);
94              for(int j=L[i];j!=i;j=L[j])
95                  resume(Col[j]);
96          }
97          resume(c);
98      }
99  }dlx;
100 int n,m,p;
101 void solve()
102 {
103     cin>>n>>m>>p;
104     dlx.build(p,n*m);
105     for(int i=1;i<=p;i++)
106     {
107         int x,y,_x,_y;
108         cin>>x>>y>>_x>>_y;
109         for(int I=x+1;I<=_x;I++)
```

```
110             for(int J=y+1;J<=_y;J++)
111                 dlx.insert(i,(I-1)*m+J);
112     }
113     dlx.ans=inf;
114     dlx.dance(0);
115     if(dlx.ans==inf)cout<<-1<<endl;
116     else cout<<dlx.ans<<endl;
117 }
118 signed main(){
119     ios::sync_with_stdio(false);
120     cin.tie(nullptr);cout.tie(nullptr);
121     int __;cin>>__;
122     while(__--)
123         solve();
124     return 0;
125 }
```

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;
```

```
36              rt=ch[rt][tmp];
37          }
38          val[rt]++;
39          id[x]=rt;
40      }
41      void build()
42      {
43          queue<int>q;
44          for(int i=0;i<26;i++)
45              if(ch[0][i])
46                  q.push(ch[0][i]);
47          while(!q.empty()){
48              int x=q.front();q.pop();
49              for(int i=0;i<26;i++){
50                  int &rt=ch[x][i];
51                  if(!rt)
52                      rt=ch[nxt[x]][i];
53                  else{
54                      nxt[rt]=ch[nxt[x]][i];
55                      in[ch[nxt[x]][i]]++;
56                      q.push(rt);
57                  }
58              }
59          }
60      }
61      void query(const string&s,int n)
62      {
63          int rt=0;
64          for(int i=0;i<s.size();i++)
65          {
66              int tmp=s[i]-'a';
67              rt=ch[rt][tmp];
68              cnt[rt]++;
69          }
70          queue<int>q;
71          for(int i=1;i<=indx;i++)
72              if(!in[i])q.push(i);
```

```
73          while(!q.empty())
74          {
75              int x=q.front();q.pop();
76              int y=nxt[x];
77              cnt[y]+=cnt[x];
78              in[y]--;
79              if(!in[y])q.push(y);
80          }
81          for(int i=1;i<=n;i++)
82              cout<<cnt[id[i]]<<endl;
83      }
84  }ac;
85  int n;
86  void solve()
87  {
88      cin>>n;
89      for(int i=1;i<=n;i++){
90          string str;cin>>str;
91          ac.insert(str,i);
92      }
93      string t;
94      cin>>t;
95      ac.build();
96      ac.query(t,n);
97  }
98  signed main(){
99   // freopen("data.in","r",stdin);
100  // freopen("data.out","w",stdout);
101     ios::sync_with_stdio(false);
102     cin.tie(nullptr);cout.tie(nullptr);
103     // int __;cin>>__;
104     // while(__--)
105         solve();
106     return 0;
107 }
```

3 HASH

```
1  /*
2  双哈希 测试
3  https://ac.nowcoder.com/acm/contest/64384/D
4  */
5  #pragma GCC optimize(2)
6  #pragma GCC optimize(3,"Ofast","inline")
7  #include<bits/stdc++.h>
8  using namespace std;
9  #define endl "\n"
10 using ull=unsigned long long;
11 using ll=long long;
12 using pii=pair<int,int>;
13 using PLL=pair<ull,ull>;
14 const int maxn=1e6+200;
15 char s[maxn];
16 int n,m,k;
17 namespace EA{
18     class Hash{
19     public:
20         ull Prime_Pool[3] = {233317ull,1998585857ul
               ,23333333333ul};
21         ull Seed_Pool[4]={911,146527,19260817,91815541};
22         ull Mod_Pool
               [5]={29123,998244353,1000000009,4294967291ull
               ,212370440130137957ll};
23         ull sum[maxn],bas[maxn];
24         int perm[maxn];
25         int sigma;
26         ull Seed,Mod;
27         int N;
28         Hash(int x,int y):Seed(Seed_Pool[x]),Mod(Mod_Pool[
               y]){
29             bas[0]=1;
30             for (int i = 1; i <= N; i++){
31                 bas[i] = bas[i - 1] * Seed % Mod;
```

```
32              }
33          }
34          void init(char *s){
35              N=strlen(s+1);
36              for(int i=1;i<=N;i++)
37                  sum[i]=(sum[i-1]*Seed%Mod+s[i])%Mod;
38              bas[0]=1;
39              for(int i=1;i<=N;i++)
40                  bas[i]=bas[i-1]*Seed%Mod;
41          }
42          void indexInit(char *s){
43              N=strlen(s+1);
44              sigma=N+100;
45              iota(perm+1,perm+1+sigma,1);
46              random_shuffle(perm+1,perm+1+sigma);
47              // for(int i=1;i<=N;i++)
48              //  cout<<perm[i]<<" ";
49              // cout<<endl;
50              for(int i=1;i<=N;i++)
51                  sum[i]=(sum[i-1]*Seed%Mod+perm[s[i]])%Mod;
52              bas[0]=1;
53              for(int i=1;i<=N;i++)
54                  bas[i]=bas[i-1]*Seed%Mod;
55          }
56          ull get(int l,int r){
57              return (sum[r]-sum[l-1]*bas[r-l+1]%Mod+Mod)%
                    Mod;
58          }
59      }h[2]{Hash(2,2),Hash(3,3)};
60      void solve()
61      {
62          cin>>n>>m>>k>>s+1;
63          h[0].init(s);
64          h[1].init(s);
65          // h[0].indexInit(s);
66          // h[1].indexInit(s);
67          map<PLL,int>cnt;
```

```
68          map<PLL,int>vis;
69          for(int i=m;i<=n;i++)
70          {
71              ull now[2];
72              now[0]=h[0].get(i-m+1,i);
73              now[1]=h[1].get(i-m+1,i);
74              PLL fl={now[0],now[1]};
75              if(vis.count(fl)&&vis[fl]<i-m+1)
76              {
77                  cnt[fl]++;
78                  vis[fl]=i;
79              }
80              else if(!vis.count(fl))
81              {
82                  cnt[fl]=1;
83                  vis[fl]=i;
84              }
85          }
86          int ans=0;
87          for(auto [x,y]:cnt)
88              if(y==k)ans++;
89          cout<<ans<<endl;
90      }
91  };
92  namespace EB{
93      class StringHash{
94      public:
95          ull Mod_Pool
               [5]={29123,998244353,1000000009,4294967291ull
               ,2123704401301379571l};
96          ull Prime_Pool[3] = {233317ull,1998585857ul
               ,23333333333ul};
97          ull seed[4]={911,146527,19260817,91815541};
98          ull mod=2123704401301379571l;
99          ull prime=233317;
100         ull base=131;
101         vector<ull>h;
```

```
102         vector<ull>bas;
103         StringHash():h(1),bas(1,1),base(131){}
104         StringHash(int x,int y):h(1),bas(1,1){base=seed[x
              ];mod=Mod_Pool[y];}
105         void push_back(char ch){
106             h.push_back((h.back()*base+ch)%mod + prime);
107             bas.push_back(bas.back()*base%mod);
108         }
109         // ull get(int l,int r){
110         //  return (h[r] + __int128(h[l])*(mod-bas[r-l]))%
              mod;
111         // }
112         ull get(int l,int r){
113             return (h[r]-h[l-1]*bas[r-l+1]%mod+mod)%mod;
114         }
115     };
116     void A()
117     {
118         string str;
119         StringHash hs,rhs;
120         int N=int(str.size());
121         for(int i=0;i<N;i++)
122             hs.push_back(str[i]);
123         for(int i=N-1;i>=0;i--)
124             rhs.push_back(str[i]);
125     }
126     void solve()
127     {
128         cin>>n>>m>>k>>s+1;
129         StringHash h[2]{StringHash(1,2),StringHash(2,3)};
130         for(int i=1;i<=n;i++)
131         {
132             h[0].push_back(s[i]);
133             h[1].push_back(s[i]);
134         }
135         map<PLL,int>cnt;
136         map<PLL,int>vis;
```

```
137         for(int i=m;i<=n;i++)
138         {
139             ull now[2];
140             now[0]=h[0].get(i-m+1,i);
141             now[1]=h[1].get(i-m+1,i);
142             PLL fl={now[0],now[1]};
143             if(vis.count(fl)&&vis[fl]<i-m+1)
144             {
145                 cnt[fl]++;
146                 vis[fl]=i;
147             }
148             else if(!vis.count(fl))
149             {
150                 cnt[fl]=1;
151                 vis[fl]=i;
152             }
153         }
154         int ans=0;
155         for(auto [x,y]:cnt)
156             if(y==k)ans++;
157         cout<<ans<<endl;
158     }
159 };
160 namespace shuang{
161     class DoubleHash{
162     public:
163         const ull b1=137,b2=149,i1=1'603'801'661,i2=1'024'
              053'074;
164         const ull p1=2'034'452'107,p2=2'013'074'419;
165         ull m1[maxn],m2[maxn],r1,r2;
166         PLL h[maxn];
167         int N;
168         void init(char*s){
169             h[0]={0,0};
170             int n=strlen(s+1);
171             m1[0]=m2[0]=1;
172             N=n+10;
```

```
173             for(int i=1;i<=N;i++)
174             {
175                 m1[i]=m1[i-1]*b1%p1;
176                 m2[i]=m2[i-1]*b2%p2;
177             }
178             h[0]={0,0};
179             for(int i=1;i<=n;i++)
180             {
181                 r1=(r1+s[i]*m1[i])%p1;
182                 r2=(r2+s[i]*m2[i])%p2;
183                 h[i]={r1,r2};
184             }
185         }
186         PLL getv(int l,int r){
187             return {(p1+h[r].first-h[l].first)*m1[N-l]%p1
                    ,(p2+h[r].second-h[l].second)*m2[N-l]%p2};
188         }
189         ull get(int l,int r){
190             PLL a=getv(l,r);
191             return (ull)a.first<<32|a.second;
192         }
193     }h;
194 };
195 signed main()
196 {
197     ios::sync_with_stdio(false);
198     cin.tie(nullptr);cout.tie(nullptr);
199     EA::solve();
200     return 0;
201 }
```

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;
```

```
36          else{
37              int q=ch[p][c];
38              if(len[q]==len[p]+1)fa[np]=q;
39              else{
40                  int nq=++tot;
41                  len[nq]=len[p]+1;
42                  fa[nq]=fa[q]; fa[q]=nq; fa[np]=nq;
43                  for(;p&&ch[p][c]==q;p=fa[p])ch[p][c]=nq;
44                  memcpy(ch[nq],ch[q],sizeof(ch[q]));
45              }
46          }
47      }
48      void insert(const char*s){for(int i=0;s[i]!='\0';i++)
          insert(s[i]-'a');}
49      void insert(const string&s){for(int i=0;i<(int)s.size
          ();i++)insert(s[i]-'a');}
50      void build_tree(){
51          for(int i=2;i<=tot;i++)
52              v[fa[i]].push_back(i);
53      }
54      /*  op=0本质不同子串 op=1位置不同子串  */
55      void build(int op=0){
56          build_tree();
57          if(!op)
58          {
59              for(int i=1;i<=tot;i++)
60                  f[i]=siz[i]=1;
61          }
62          else get_siz(1);
63          f[1]=siz[1]=0;
64          get_f(1);
65      }
66      void get_kth(int k,int rt=1)
67      {
68          if(k>f[rt]){
69              cout<<-1<<endl;
70              return ;
```

```
71          }
72          if(k<=siz[rt]){cout<<"\n";return ;}
73          k-=siz[rt];
74          if(k==0){return;}
75          for(int i=0;i<26;i++){
76              if(ch[rt][i]){
77                  if(f[ch[rt][i]]>=k){
78                      cout<<(char)(i+'a');
79                      get_kth(k,ch[rt][i]);
80                      return ;
81                  }
82                  else
83                      k-=f[ch[rt][i]];
84              }
85          }
86          cout<<-1<<endl;
87      }
88      void match(const string&s)
89      {
90          // queue<int>q;
91          // for(int i=1;i<=tot;i++){
92          //  f[i]=0,in[i]=main_in[i];
93          //  if(!in[i])q.push(i);
94          // }
95          // int rt=1;
96          // int length=0;
97          // for(int i=0,tmp;i<(int)s.size();i++)
98          // {
99          //  tmp=s[i]-'a';
100         //  while(rt&&!ch[rt][tmp])
101         //      rt=fa[rt],length=len[rt];
102         //  if(!rt)rt=1;
103         //  else{
104         //      rt=ch[rt][tmp];length++;
105         //  }
106         //  f[rt]=max(f[rt],length);
107         // }
```

```
108         // while(!q.empty()){
109         //  int x=q.front();q.pop();
110         //  mi[x]=min(mi[x],f[x]);
111         //  int y=fa[x];
112         //  if(f[x]>=len[y])
113         //      f[y]=len[y];
114         //  in[y]--;
115         //  if(!in[y])q.push(y);
116         // }
117     }
118     void debug(){
119         for(int i=0;i<=tot;i++)
120             for(int j=0;j<26;j++)
121                 if(ch[i][j])
122                     cout<<"ins␣"<<i<<"␣--"<<(char)(j+'a')
                            <<"-->␣"<<ch[i][j]<<endl;
123         for(int i=1;i<=tot;i++)
124             cout<<"fail␣"<<i<<"␣->␣"<<fa[i]<<endl;
125         for(int i=1;i<=tot;i++)
126             cout<<"len["<<i<<"]="<<len[i]<<endl;
127     }
128     void query();
129 private:
130     void get_siz(int x){
131         for(auto y:v[x]){
132             get_siz(y);
133             siz[x]+=siz[y];
134         }
135         f[x]=siz[x];
136     }
137     void get_f(int x){
138         if(vis[x])return ;
139         vis[x]=1;
140         for(int i=0;i<26;i++){
141             int y=ch[x][i];
142             if(!y)continue;
143             get_f(y);
```

```
144             f[x]+=f[y];
145         }
146     }
147 }sam;
148 char ch[maxn];
149 int op,k;
150 void solve()
151 {
152     cin>>ch;
153     sam.insert(ch);
154     // sam.debug();
155     sam.build(0);
156     int q;cin>>q;
157     while(q--){
158         cin>>k;
159         sam.get_kth(k);
160     }
161 }
162 signed main(){
163  // freopen("data.in","r",stdin);
164  // freopen("data.out","w",stdout);
165     ios::sync_with_stdio(false);
166     cin.tie(nullptr);cout.tie(nullptr);
167         solve();
168     return 0;
169 }
```

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;
```

```
36      }
37      int get_fail(int rt){
38          while(s[now-len[rt]-2]!=s[now-1])rt=fa[rt];
39          return rt;
40      }
41      void insert(int c)
42      {
43          s[now++]=c;
44          int rt=get_fail(last);
45          if(!ch[rt][c]){
46              int np=newnode(len[rt]+2);
47              fa[np]=ch[get_fail(fa[rt])][c];
48              ch[rt][c]=np;
49              // num[np]=num[fa[np]]+1;
50          }
51          last=ch[rt][c];
52          num[last]++;
53      }
54      void build(){
55          for(int i=tot;i>=2;i--)
56              num[fa[i]]+=num[i];
57          num[0]=num[1]=0;
58      }
59      void insert(char*s){while(*s){insert(*s-'a');s++;}}
60      void insert(const string&s){for(auto i:s)insert(i-'a')
            ;}
61      void debug(){
62          for(int i=0;i<=tot;i++)
63              for(int j=0;j<26;j++)
64                  if(ch[i][j])
65                      cout<<"ins␣"<<i<<"␣--"<<(char)(j+'a')
                            <<"-->␣"<<ch[i][j]<<endl;
66          for(int i=0;i<=tot;i++)
67              cout<<"len["<<i<<"]="<<len[i]<<endl;
68          for(int i=0;i<=tot;i++)
69              cout<<"fa["<<i<<"]="<<fa[i]<<endl;
70      }
```

```
71 }pam;
72 string s;
73 void solve()
74 {
75     cin>>s;
76     pam.insert(s);
77 }
78 signed main(){
79  // freopen("data.in","r",stdin);
80  // freopen("data.out","w",stdout);
81     ios::sync_with_stdio(false);
82     cin.tie(nullptr);cout.tie(nullptr);
83         solve();
84     return 0;
85 }
```

6 倍增优化建图

```
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);
```

```
36          int lca=LCA(x,y);
37          for(int i=14;i>=0;i--)
38              if((dep[x]-dep[lca])>>i&1)
39              {
40                  Flow::Add(now,find(x,i),INF);
41                  x=f[x][i];
42              }
43          for(int i=14;i>=0;i--)
44              if((dep[y]-dep[lca])>>i&1)
45              {
46                  Flow::Add(now,find(y,i),INF);
47                  y=f[y][i];
48              }
49      }
50      ans-=Flow::DINIC();
51      cout<<ans<<endl;
52  }
```

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)
{
```

```
36      dis[++cnt]=d[x];
37      for(int i=head[x];i;i=edge[i].next)
38      {
39          int y=edge[i].to;
40          if(y==fa||del[y])continue;
41          d[y]=d[x]+edge[i].w;
42          getdis(y,x);
43      }
44  }
45  void calc(int x,int w,int op)
46  {
47      cnt=0,d[x]=w;
48      getdis(x,0);
49      sort(dis+1,dis+1+cnt);
50      for(int i=1;i<=m;i++)
51      {
52          int l=1,r=cnt;
53          while(l<r){
54              if(dis[l]+dis[r]<=ask[i]){
55                  if(dis[l]+dis[r]==ask[i])ans[i]+=op;
56                  ++l;
57              }
58              else --r;
59          }
60      }
61  }
62  void divide(int x)
63  {
64      calc(x,0,1);
65      del[x]=1;
66      for(int i=head[x];i;i=edge[i].next)
67      {
68          int y=edge[i].to;
69          if(del[y])continue;
70          calc(y,edge[i].w,-1);
71          maxson=sum=siz[y];
72          getroot(y,0);
```

```
73          divide(root);
74      }
75  }
76  char ch[maxn];
77  void solve()
78  {
79      cin>>n>>m;
80      for(int i=1;i<n;i++)
81      {
82          int x,y,w;
83          cin>>x>>y>>w;
84          add(x,y,w);
85          add(y,x,w);
86      }
87      for(int i=1;i<=m;i++)
88          cin>>ask[i];
89      maxson=sum=n;
90      getroot(1,0);
91      getroot(root,0);
92      divide(root);
93      for(int i=1;i<=m;i++)
94          cout<<(ans[i]?"AYE\n":"NAY\n");
95  }
96  signed main(){
97      ios::sync_with_stdio(false);
98      cin.tie(nullptr);cout.tie(nullptr);
99          solve();
100     return 0;
101 }
```

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;
}
```

```
36 void add(int from,int to,int w)
37 {
38     edge[++cnt].w=w;
39     edge[cnt].to=to;
40     edge[cnt].next=head[from];
41     head[from]=cnt;
42 }
43 void tarjan(int x,int fa)
44 {
45     dfn[x]=low[x]=++indx;
46     for(int i=head[x];i;i=edge[i].next)
47     {
48         int y=edge[i].to;
49         if(!dfn[y])
50         {
51             s.push({x,edge[i].w});
52             tarjan(y,i);
53             low[x]=min(low[x],low[y]);
54             if(low[y]>=dfn[x])
55             {
56                 vector<int>g;
57                 pii v;
58                 do{
59                     v=s.top();s.pop();
60                     g.push_back(v.second);
61                 }while(v.first!=x);
62 
63                 sort(g.begin(),g.end());
64 
65                 if(g.size()==1)ans=min(ans,g[0]);
66                 else if(g.size()==2)ans=min(ans,g[0]+g[1])
                    ;
67                 else ans=min(ans,min(g[0]+g[1],g[2]));
68             }
69         }
70         else if(dfn[x]>dfn[y]&&i!=(fa^1))
71         {
```

```
72              low[x]=min(low[x],dfn[y]);
73              s.push({x,edge[i].w});
74          }
75      }
76  }
77  void solve()
78  {
79      cin>>n>>m;
80      init();
81      for(int i=1;i<=m;i++)
82      {
83          int x,y,w;
84          cin>>x>>y>>w;
85          add(x,y,w);
86          add(y,x,w);
87      }
88      tarjan(1,0);
89      cout<<ans<<endl;
90  }
91  signed main(){
92      int size(512<<20); // 512M
93      __asm__ ( "movq␣%0,␣%%rsp\n"::"r"((char*)malloc(size)+
            size)); // YOUR CODE
94      ios::sync_with_stdio(false);
95      cin.tie(nullptr);cout.tie(nullptr);
96      int __;cin>>__;
97      while(__--)
98          solve();
99      exit(0);
100 }
```

9 DLX