SAJIHPTS Code Snippets

SAJIHPTS Team

May 13, 2017

Contents

1 Template

1.1 Aho_Corasick_Automata

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
#include<queue>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}

#define MAXNode (1000000)
#define Sigma_size (26)
class Aho_Corasick_Automata
{
public:
  int ch[MAXNode][Sigma_size];
  int v[MAXNode],siz;
  // AC 自动机
  int f[MAXNode],last[MAXNode];
  Aho_Corasick_Automata(int _siz=0):siz(_siz){MEM(ch) MEM(v)
  ↪ MEM(f) MEM(last)}
```

```
42    void mem(int _siz=0){siz=_siz; MEM(ch) MEM(v) MEM(f)
      ↪  MEM(last)  }
43    int idx(char c){return c-'a';}
44    void insert(char *s,int val=1) //val 不为 0 表示 str 末尾
45    {
46      int u=0,n=strlen(s);
47      Rep(i,n)
48      {
49        int c=idx(s[i]);
50        if (!ch[u][c])
51        {
52          ++siz;
53          MEM(ch[siz]);
54          ch[u][c]=siz;
55        }
56        u=ch[u][c];
57      }
58      v[u]=val;
59    }
60    void getFail()
61    {
62      queue<int> q;
63      Rep(c,Sigma_size)
64      {
65        int u=ch[0][c];
66        if (u) q.push(u),last[u]=0;
67      }
68      while (!q.empty())
69      {
70        int r=q.front();q.pop();  //r--c-->u
71        Rep(c,Sigma_size)
72        {
73          int u=ch[r][c];
74          if (!u) {ch[r][c]=ch[f[r]][c]; continue;}
75          q.push(u);
76          f[u]=ch[f[r]][c];
77          last[u]=v[f[u]]?f[u]:last[f[u]];
78        }
79      }
80    }
81    void print(int j) //打印全串中所有以 j 为末尾的 str
82    {
83      if (j)
84      {
85        printf("%d %d\n",j,v[j]);
86        print(last[j]);
```

```
    }
  }
  void find(char *s)
  {
    int u=0,n=strlen(s);
    Rep(i,n)
    {
      int c=idx(s[i]);
      u=ch[u][c];
      if (v[u]) print(u);
      else if (last[u]) print(u);
    }
  }

}T;

int main()
{
  freopen(".in","r",stdin);
//  freopen(".out","w",stdout);



  return 0;
}
```

”

1.2 Bellman_Ford

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case %d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
typedef long long ll;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
int n,m;
```

```cpp
struct Edge{
  int from,to;
  double dist;
};
#define MAXN (1000)
struct BellmanFord {
  int n,m;
  vector<Edge> edges;
  vi G[MAXN];
  bool inq[MAXN];
  double d[MAXN];
  int cnt[MAXN],p[MAXN];
  void addedge(int u,int v,int w){
    edges.pb((Edge){u,v,w});
    G[u].pb(m++);
  }
  void addedge2(int u,int v,int w) {
    addedge(u,v,w);addedge(v,u,w);
  }
  void init(int _n){
    n=_n; m = 0;
    Rep(i,n) G[i].clear();
    edges.clear();
  }
  bool negativeCycle() {
    queue<int> Q;
    MEM(inq) MEM(cnt)
    Rep(i,n) d[i]=0,inq[i]=1,Q.push(i);
    while(!Q.empty()) {
      int u = Q.front(); Q.pop();
      inq[u] = 0;
      int mm=G[u].size();
      Rep(i,mm) {
        Edge e = edges[G[u][i]];
        if (d[e.to]>d[u]+e.dist ) {
          d[e.to]=d[u]+e.dist;
          p[e.to]=G[u][i];

          if (!inq[e.to]) {
            Q.push(e.to);
            inq[e.to]=1;
            if (++cnt[e.to]>n) return 1;
          }
        }
      }
    }
```

```
91      return 0;
92    }
93
94  }S1;
95
96  int main()
97  {
98  //  freopen(".in","r",stdin);
99  //  freopen(".out","w",stdout);
100
101   return 0;
102 }
```

”

1.3 BigInt Alpha0

```cpp
//#include "stdafx.h"
#include <iostream>
#include <cmath>
#include <vector>
#include <cstdio>
#include <string>
#include <algorithm>
//#pragma warning(disable:4996)

using namespace std;

#define LL long long
#define ULL unsigned long long
#define LD long double

#define Rep(x,y,i) for (int i=x;i<y;i++) //[x,y)
#define RepD(x,y,i) for (int i=x;i>y;i--) //(y,x]
#define Mem(X) memset(X,0,sizeof(X));
#define Pr(X) cout<<" "<<#X<<"="<<X<<" ";
#define PrL(X) cout<<#X<<" = "<<X<<endl;
#define PrLL cout<<endl;
using namespace std;
const double EPS = 1e-10;


#define IntMod 10000
struct BigInt
{
private:
  vector<int> A;
  bool Positive;
  int VecNum;

public:
  //Constructors
  inline int GetLength(LL a) {int t=0; while (a>0) {a/=IntMod;
  ↪  t++;} return t;}
  BigInt() { Positive = 1; VecNum = 0; }
  BigInt(const BigInt &a) { A = a.A; VecNum = a.VecNum; Positive =
  ↪  a.Positive; }
  BigInt(string s)
  {
    A.reserve(100);
```

```
    if (s == "-0") { A.push_back(0); Positive = 1; VecNum = 1;
    ↪  return; }
    int kk = 3; int kl = 0;
    int l = s.length();
    int j = l - 1;
    if (s[0] == '-') { Positive = 0; kk++; kl++; }
    else Positive = 1;
    while (j>kk)
    {
      int t = 0;
      Rep(0, 4, i) { t *= 10; t += s[j - (3 - i)] - '0'; }
      j -= 4;
      A.push_back(t);
    }
    int t = 0;
    int k = j + 1;
    Rep(kl, k, i) { t *= 10; t += s[i] - '0'; }
    A.push_back(t);
    VecNum = A.size();
  }
  BigInt(const LL &b)
  {
    LL a=b;
    Positive=(a>=0);
    VecNum=GetLength(abs(a));
    A.resize(VecNum);
    Rep(0,VecNum,i)
    {
      A[i]=a%IntMod;
      a/=IntMod;
    }
  }
  //BigInt& operator = (BigInt& a) { VecNum=a.VecNum;
  ↪  Positive=a.Positive; A=a.A; }
  BigInt& operator = (const string s) { BigInt x(s); *this=x; }
  BigInt& operator = (LL s) { BigInt x(s); *this = x; }

  //Basic Math Functions
  friend BigInt abs(BigInt a) { a.Positive=1; return a; }
  friend bool isnegative(BigInt &a) { return a.Positive; }
  BigInt& operator - () { Positive = !(Positive); return *this; }

  // Ostream and Instream
  friend ostream& operator << (ostream &out, BigInt &a)
  {
    if (a.VecNum==0)
```

```
86      {
87        out<<"0";
88        return out;
89      }   //Bug Fixed if there is a BigInt constructed by the
        ↪  default constructor
90      if (!a.Positive) out << "-";
91      out << a.A[a.VecNum - 1];
92      RepD(a.VecNum - 2, -1, i)
93      {
94        if (a.A[i] == 0) { out << "0000"; continue; }
95        Rep(0, (int)(4 - log(a.A[i]) / log(10) - EPS), j) out <<
          ↪  '0';
96        out << a.A[i];
97      }
98      return out;
99    }
100   friend istream& operator >> (istream &in, BigInt &a)
101   {
102     string s;
103     in >> s;
104     int L = s.length() - 1;int i = 0;bool flag = 0;
105     if (s[i] == '-') { i++; flag = 1; }
106     while (s[i] == '0' && i < L) i++;
107     string b(s.begin() + i, s.end());
108     if (flag) b.insert(0, "-");
109     a = BigInt(b);
110     return in;
111   }
112
113   //Bool Operators
114   bool operator < (const BigInt &b) const
115   {
116     if (Positive &&  b.Positive)
117     {
118       if (VecNum != b.VecNum) return (bool)(VecNum<b.VecNum);
119       RepD(VecNum-1, -1, i)
120         if (A[i] != b.A[i])
121           return (bool)(A[i]<b.A[i]);
122       //Bug Fixed that there should be a RepD rather that Rep
123       return 0;
124     }
125     if (!Positive &&  b.Positive) return 1;
126     if (Positive && !b.Positive)  return 0;
127     BigInt a = b;
128     BigInt c = (*this);
129     if (!c.Positive && !a.Positive) return !((-c) < (-a));
```

```
130       return 1;
131     }
132     bool operator >  (const BigInt &b) const { return !((*this)<b);
        ↪  }
133     bool operator == (const BigInt &b) const { return (Positive ==
        ↪  b.Positive && A == b.A && VecNum == b.VecNum); }
134     bool operator <= (const BigInt &b) const { return (*this == b)
        ↪  || (*this < b); }
135     bool operator >= (const BigInt &b) const { return (*this == b)
        ↪  || !(*this < b);}
136     bool operator != (const BigInt &b) const { return !(*this == b);
        ↪  }
137
138     bool operator <  (const string b) const { BigInt x(b); return
        ↪  (*this) < x; }
139     bool operator == (const string b) const { BigInt x(b); return
        ↪  x==(*this) < x; }
140     bool operator >  (const string b) const
141     bool operator <= (const string b) const
142     bool operator >= (const string b) const
143     bool operator != (const string b) const
144
145     bool operator <  (const LL& b) const { BigInt x(b); return *this
        ↪  < x; }
146     bool operator == (const LL& b) const { BigInt x(b); return *this
        ↪  == x; }
147     bool operator >  (const LL& b) const
148     bool operator <= (const LL& b) const
149     bool operator >= (const LL& b) const
150     bool operator != (const LL& b) const
151
152     //Function of Plus and Minus
153     BigInt operator - (const BigInt &b) const
154     {
155       BigInt x=*this;
156       BigInt y=b;
157
158       if (!x.Positive &&  y.Positive) return -(-x + y);
159       if ( x.Positive && !y.Positive) return x + y;
160       if (!x.Positive && !y.Positive) return (-y) - (-x);
161       if (x<y) return  -(y - x);
162
163       int L = max(x.VecNum, y.VecNum);
164       y.A.resize(L);
165       x.A.resize(L);
166       Rep(0, L, i)
```

```
167       {
168         x.A[i] -= y.A[i];
169         if (x.A[i]<0) { x.A[i] += IntMod; x.A[i + 1]--; }
170       }
171       while (x.A[L - 1] == 0 && ((L-1)!=0) ) { x.VecNum--;
          ↪  x.A.pop_back(); L--; }
172       return x;
173     }
174     BigInt operator + (const BigInt &b) const
175     {
176       BigInt x=*this;
177       BigInt y=b;
178
179       if (!x.Positive &&  y.Positive) return y - (-x);
180       if (!x.Positive && !y.Positive) return -(-x + (-y));
181       if ( x.Positive && !y.Positive)  return x - (-y);
182       int L = max(x.VecNum, y.VecNum);
183       x.A.resize(L + 1);
184       y.A.resize(L + 1);
185       for (int i = 0; i<L; i++) x.A[i] += y.A[i];
186       for (int i = 0; i<L; i++)
187       { x.A[i + 1] += x.A[i] / IntMod; x.A[i] %= IntMod; }
188       x.VecNum = L;
189       if (x.A[L]) x.VecNum++;
190       else x.A.erase(x.A.begin() + L);
191       return x;
192     }
193     BigInt operator - (const LL &b) const { BigInt y(b); return
          ↪  *this - y; }
194     BigInt operator + (const LL &b) const { BigInt y(b); return
          ↪  *this + y; }
195     BigInt operator - (const string b) const { BigInt y(b); return
          ↪  *this - y; }
196     BigInt operator + (const string b) const { BigInt y(b); return
          ↪  *this + y; }
197     BigInt operator += (const BigInt& b) { *this=*this+b; return
          ↪  *this; }
198     BigInt operator += (const LL& b)     { *this=*this+b; return
          ↪  *this; }
199     BigInt operator -= (const BigInt& b) { *this=*this-b; return
          ↪  *this; }
200     BigInt operator -= (const LL& b)     { *this=*this-b; return
          ↪  *this; }
201
202
203     //Function of Multiply and Division
```

```
204   BigInt operator * (const BigInt& b) const
205   {
206     BigInt x=*this;
207     BigInt y=b;
208     BigInt ans;
209     if ((x.Positive && y.Positive) || (!x.Positive &&
      ↪  !y.Positive)) ans.Positive=1;
210     else ans.Positive=0;
211     int m=x.VecNum,n=y.VecNum;
212     int L=m+n+1;
213     ans.VecNum=L;
214     ans.A.resize(L+3);
215     Rep(0,m,i)
216       Rep(0,n,j)
217       {
218         int pos=i+j;
219         LL t=x.A[i]*y.A[j];
220         int post=0;
221         while (t>0)
222         {
223           ans.A[pos+post]+=t%IntMod;
224           t/=IntMod;
225           post++;
226         }
227       }
228     Rep(0,L,i) { ans.A[i + 1] += ans.A[i] / IntMod; ans.A[i] %=
      ↪  IntMod; }
229     while (ans.A[L - 1] == 0 && ((L-1)!=0) ) { ans.VecNum--;
      ↪  ans.A.pop_back(); L--; }
230     return ans;
231   }
232   BigInt operator * (const LL &b) const    { BigInt x(b); return
      ↪  (*this) * x; }
233   BigInt operator * (const string b) const { BigInt x(b); return
      ↪  (*this) * x; }
234   BigInt operator *= (const BigInt& b){ *this = *this * b; return
      ↪  *this; }
235   BigInt operator *= (const LL &b)    { *this = *this * b; return
      ↪  *this; }
236   BigInt operator *= (const string b) { *this = *this * b; return
      ↪  *this; }
237
238   BigInt operator / (const BigInt& b) const
239   {
240
241   }
```

```
242
243   //Function of Remainder
244   BigInt operator % (const BigInt& b) const
245   {
246
247   }
248
249   //Function of BITS
250 };
251
252 int main()
253 {
254   //freopen("D:\\YA.in","r",stdin);
255   //freopen("mul.in","r",stdin);
256   //freopen("mul.out","w",stdout);
257   BigInt a(123456);
258
259   BigInt b=(string)"123456";
260
261   BigInt c=a*"123456";
262   cout<<c<<endl;
263   //BigInt c=a*b;
264   //cout<<c<<endl;
265 }
266
267 //Equals A/B
268 struct BigDec:public BigInt
269 {
270 private:
271   BigInt A;
272   BigInt B;
273   bool Positive;
274
275 public:
276   BigDec() { Positive=1; A="0"; B="1"; }
277   BigDec(string s) { }
278   BigDec(double t) { }
279   BigDec(LL t) { }
280 };
```

”

1.4 BigInteger

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,0x3f,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEMx(a,b) memset(a,b,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
#define ALL(x) (x).begin(),(x).end()
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
```

```
45    while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
46    return x*f;
47  }
48  int n,k;
49
50  #define MAXN (10000)
51  struct BigInteger {
52    int n;
53    int a[MAXN];
54    enum {MOD=10000};
55    BigInteger() {
56      MEM(a) a[0]=1;
57    }
58      int& operator [] (int p) {return a[p];}
59      const int& operator [] (int p) const {return a[p];}
60    BigInteger(int x) {
61      a[0]=1;
62      a[1]=x%MOD;
63      x/=MOD;
64      while(x) a[++a[0]]=x%MOD,x/=MOD;
65    }
66    friend BigInteger operator*(BigInteger a,BigInteger b) {
67      BigInteger c;
68      c[0]=a[0]+b[0];
69      For(i,a[0])
70              For(j,b[0]){
71                  c[i+j-1]+=a[i]*b[j], c[i+j]+=c[i+j-1]/MOD,
                        ↪  c[i+j-1]%=MOD;
72              }
73          if (c[c[0]] == 0) c[0]--;
74          return c;
75    }
76    void print() {
77        ForD(i,a[0]) {
78          if(i==a[0]) printf("%d",a[i]);
79          else printf("%04d",a[i]);
80        }
81    }
82  };
83  int gcd(int a,int b){if (!b) return a;return gcd(b,a%b);}
84  void calc() {
85    if(n>k) {
86      puts("0 1");
87      return ;
88    }
89    BigInteger a,b;
```

```
  a=1; b=1;
  For(i,n-1) a=a*(k+1);
  int p=k-n+1,w=1;
  For(i,n) {
    int g=gcd(p,k);
    p/=g;
    b=b*(k/g);
  }
  a=a*p;
  a.print();
  putchar(' ');
  b.print();
  puts("");
}
int main()
{
//  freopen("bzoj3680.in","r",stdin);
//  freopen(".out","w",stdout);
  int T=read();
  while(T--) {
    n=read(),k=read();
    calc();
  }
  return 0;
}
```

”

1.5 Binary_Index_Tree

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %I64d\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
```

```
}
int n,m;
struct BIT{
  #define MAXN (5000000+10)
  ll f[MAXN];
  void add(int x,ll v) {
    for(int i=x;i<=n;i+=i&(-i))
         f[i]+=v;
  }
  ll qur(int x) {
    ll v=0;
    for(int i=x;i;i-=i&(-i))
         v+=f[i];
    return v;
  }
}T;
ll a[MAXN];
int main()
{
//  freopen(".in","r",stdin);
//  freopen(".out","w",stdout);
  n=read();
  MEM(T.f)
  For(i,n) {
    a[i]=read();
  }
  while(m--) {
  }
  return 0;
}
```

”

1.6 Binary_Index_Tree2D

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %I64d\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
```

```
}
int n,m;
#define MAXN (2000+10)
ll f[MAXN][MAXN]={0};
void add(int x,int y,ll v) {
  for(int i=x;i<=n;i+=i&(-i))
    for(int j=y;j<=m;j+=j&(-j))
       f[i][j]+=v;
}
ll qur(int x,int y) {
  ll v=0;
  for(int i=x;i;i-=i&(-i))
    for(int j=y;j;j-=j&(-j))
       v+=f[i][j];
  return v;
}
int main()
{
//  freopen(".in","r",stdin);
//  freopen(".out","w",stdout);
  cin>>n>>m;
  return 0;
}
```

”

1.7 bingchaji

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (3*50000+10)
long long mul(long long a,long long b){return (a*b)%F;}
long long add(long long a,long long b){return (a+b)%F;}
long long sub(long long a,long long b){return
 ↪  (a-b+(a-b)/F*F+F)%F;}
typedef long long ll;
class bingchaji
{
public:
  int father[MAXN],n,cnt;
  void mem(int _n)
  {
    n=cnt=_n;
    For(i,n) father[i]=i;
  }
  int getfather(int x)
  {
    if (father[x]==x) return x;

    return father[x]=getfather(father[x]);
  }
```

```
44    void unite(int x,int y)
45    {
46      x=getfather(x);
47      y=getfather(y);
48      if (x^y) {
49        --cnt;
50        father[x]=y;
51        sz[y]+=sz[x];
52      }
53    }
54    bool same(int x,int y)
55    {
56      return getfather(x)==getfather(y);
57    }
58  }S;
59
60  int main()
61  {
62  //  freopen(".in","r",stdin);
63  //  freopen(".out","w",stdout);
64
65    return 0;
66  }
```

”

1.8 bingchaji_weighted

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,0x3f,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEMx(a,b) memset(a,b,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
```

```
  return x*f;
}
#define MAXN (500000  +10)
int n,m;
int dis[MAXN],father[MAXN];
int getfa(int i) {
  if (father[i]!=i && getfa(father[i])) {
    dis[i]+=dis[father[i]];
  }
  return father[i]=father[father[i]];
}
int main()
{
//  freopen(".in","r",stdin);
//  freopen(".out","w",stdout);

  while(scanf("%d%d",&n,&m)) {
    if (!n&&!m) return 0;
    For(i,n) father[i]=i,dis[i]=0;
    For(i,m) {
      int op=read(),a=read(),b=read();
      int fa=getfa(a),fb=getfa(b);
      if (op==0) {
        if (fa^fb) {
          father[fa]=fb; dis[fa]=read()+dis[b]-dis[a];
        } else read();
      } else {
        if (fa^fb) puts("UNKNOWN"); else
         ↪  printf("%d\n",dis[a]-dis[b]);
      }
    }
  }


  return 0;
}
```

”

1.9 Block_tree

```
#include<bits/stdc++.h>
using namespace std;
#define ALL(x) (x).begin(),(x).end()
const int N=200011;
inline int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
int Pre[N],Edge[N<<1],Next[N<<1];
int PRE[N],EDGE[N<<1],NEXT[N<<1];
int a[N],fa[N],belong[N];
int n,m,cnt,ans,CNT,tot,B;
struct node
{
    int size;
    vector<int> a;
    inline void insert(int x)
    {
      ++size;
        a.insert(lower_bound(ALL(a),x+1),x);
    }
    inline void change(int x,int y)
    {
      del(x); insert(y);
  }
    inline void del(int x)
    {
      --size;
      a.erase(lower_bound(ALL(a),x));
    }
    inline int query(int x)
    {
        int t=upper_bound(ALL(a),x)-a.begin()+1;
        return size-t+1;
    }
}block[200200];
inline void addedge(int x,int y)
{
    Next[++cnt]=Pre[x];
    Pre[x]=cnt;
    Edge[cnt]=y;
```

```
}
inline void deledge(int x,int y)
{
  if (Edge[Pre[x]]==y) Pre[x]=Next[Pre[x]];
  else
  for(int p=Pre[x];p;p=Next[p]) {
    if (Edge[Next[p]]==y) {
      Next[p]=Next[Next[p]];
      return;
    }
  }
}
inline void DELEDGE(int x,int y)
{
  if (EDGE[PRE[x]]==y) PRE[x]=NEXT[PRE[x]];
  else
  for(int p=PRE[x];p;p=NEXT[p]) {
    if (EDGE[NEXT[p]]==y) {
      NEXT[p]=NEXT[NEXT[p]];
      return;
    }
  }
}

inline void INSERT(int x,int y)
{
    NEXT[++CNT]=PRE[x];
    PRE[x]=CNT;
    EDGE[CNT]=y;
}
void dfs(int x)
{
    if (block[belong[fa[x]]].size==B)
        be-
         ↪  long[x]=++tot,block[tot].insert(a[x]),INSERT(belong[fa[x]],tot);
    else belong[x]=belong[fa[x]],block[belong[x]].insert(a[x]);
    for (int i=Pre[x];i;i=Next[i])
        if (Edge[i]!=fa[x])
            fa[Edge[i]]=x,dfs(Edge[i]);
}
void dfs1(int x,int y)
{
    ans+=block[x].query(y);
    for (int i=PRE[x];i;i=NEXT[i])
        dfs1(EDGE[i],y);
}
```

```
90  void query(int x,int y)
91  {
92      if (a[x]>y) ans++;
93      for (int i=Pre[x];i;i=Next[i])
94          if (Edge[i]!=fa[x])
95          {
96              if (belong[Edge[i]]==belong[x]) query(Edge[i],y);
97              else dfs1(belong[Edge[i]],y);
98          }
99  }
100
101
102 void dfs2(int x,int col,int tocol)
103 {
104   belong[x]=tocol;
105   block[tocol].insert(a[x]);
106   block[col].del(a[x]);
107
108     for (int i=Pre[x];i;i=Next[i])
109         if (Edge[i]!=fa[x]){
110           if (belong[Edge[i]]==col) {
111               dfs2(Edge[i],col,tocol);
112           } else {
113             DELEDGE(col,belong[Edge[i]]);
114             INSERT(tocol,belong[Edge[i]]);
115           }
116     }
117 }
118
119 int main()
120 {
121 //  freopen("bzoj3731.data","r",stdin);
122 //  freopen("bzoj3731.out","w",stdout);
123     n=read(); B=static_cast<int>(sqrt(n*7*log2(n))+1e-7);
124     for (int i=1;i<n;i++)
125     {
126         int u=read(),v=read();
127         addedge(u,v); addedge(v,u);
128     }
129     for (int i=1;i<=n;i++) a[i]=read();
130     dfs(1);
131     m=read();
132     for (int i=1;i<=m;i++)
133     {
134 //      ans=0;
135         int opt=read(),u=read()^ans,x;
```

```
136     if (opt<3) x=read()^ans;
137         if (opt==0)
138         {
139             ans=0;
140             if (fa[u]) query(u,x);
141             else dfs1(belong[u],x);
142             printf("%d\n",ans);
143         }
144         else if (opt==1)
145         {
146             block[belong[u]].change(a[u],x);
147             a[u]=x;
148         }
149         else if (opt==2)
150         {
151             a[++n]=x;
152             addedge(u,n);
153             fa[n]=u;
154             if (block[belong[u]].size==B)
155                 be-
                    ↪  long[n]=++tot,block[tot].insert(x),INSERT(belong[u],tot);
156             else
157             {
158                 belong[n]=belong[u],block[belong[n]].insert(a[n]);
159             }
160         }
161         else {
162           if (!fa[u]) continue;
163           int f=fa[u];
164           if (belong[f]!=belong[u]) {
165             deledge(f,u); deledge(u,f);
166             DELEDGE(belong[f],belong[u]);
167             fa[u]=0;
168           }
169           else {
170             deledge(f,u); deledge(u,f);
171         ++tot;
172             int col=belong[u];
173         dfs2(u,col,tot);
174         fa[u]=0;
175           }
176         }
177     }
178     return 0;
179 }
```

”

1.10 china

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define PRi(a,n) Rep(i,n-1) cout<<a[i]<<' '; cout<<a[n-1]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
typedef long long ll;
ll F;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
void gcd(ll a,ll b,ll &d,ll &x,ll &y) {
  if (!b) {d=a,x=1,y=0;  }
  else {gcd(b,a%b,d,y,x); y-=x*(a/b);  }
}
// x mod m0=a0,x mod m =a,noSolution return 0
```

```cpp
//初始可令 m0 = 1 ,a0 = 0
bool china(ll &m0,ll &a0,ll m,ll a)
{
  ll g,x,y;
  ll c=abs(a-a0);
  gcd(m0,m,g,x,y);
  if ( c % g ) return 0;
  x*=(a-a0)/g;
  x%=m/g;
  a0=x*m0+a0;
  m0*=m/g;
  a0%=m0;
  if(a0<0) a0+=m0;
  return 1;
}
int q1[MAXN],m1[MAXN];

int main()
{
//  freopen(".in","r",stdin);
//  freopen(".out","w",stdout);

  ll m0=1,a0=0;
  bool flag=1;
  Rep(i,n) {
    flag=china(m0,a0,m1[i],q1[i]);
    if (!flag) break;
  }
  if (flag) printf("%I64d\n",(!a0)?m0:a0);
  else puts("Creation August is a SB!");



  return 0;
}
```

”

1.11 Closest_Pair

```cpp
#include<iostream>
#include<cmath>
#include<cstdio>
#include<iomanip>
#include<algorithm>
#include<cstring>
#include<functional>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
typedef long double ll;
typedef long double ld;
typedef unsigned long long ull;
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
ld sqr(ld a){return a*a;}
#define MAXN (200000+10)
class P{
public:
```

```
45    ld x,y;
46    int id;
47    P(){}
48    P(ld x,ld y):x(x),y(y){}
49    friend long double dis2(P A,P B){return
      ↪  sqr(A.x-B.x)+sqr(A.y-B.y);  }
50    friend long double dis(P A,P B){return sqrt(dis2(A,B));  }
51    friend bool operator<(P A,P B) {
52      return A.x<B.x;
53    }
54  }a[MAXN];
55  int t[MAXN];
56  int cmp(const void *x,const void *y) {
57    return a[*(int*)x].y-a[*(int*)y].y;
58  }
59  ld bsearch(int l,int r) {
60    int m=(l+r)/2;
61    if (l==r) return 1e30;
62
63    ld d=min(bsearch(l,m),bsearch(m+1,r));
64
65    int k=0;
66    Fork(i,l,r) {
67      if (fabs(a[m].x-a[i].x)<=d) {
68        t[++k]=i;
69      }
70    }
71    qsort(t+1,k,sizeof(int),cmp);
72
73    For(i,k) {
74      Fork(j,i+1,min(k,i+7)) {
75        if (a[t[j]].y-a[t[i]].y>d) break;
76        if (a[t[i]].id!=a[t[j]].id) d=min(d,dis(a[t[i]],a[t[j]]));
77      }
78    }
79    return d;
80  }
81  int main()
82  {
83  //  freopen("Closest_Pair.in","r",stdin);
84  //  freopen(".out","w",stdout);
85    int T=read();
86    while(T--) {
87      int n=read();
88      For(i,2*n) a[i].x=read(),a[i].y=read();
89      For(i,2*n) a[i].id=i<=n;
```

```
90
91      sort(a+1,a+1+2*n);
92      cout<<setiosflags(ios::fixed)<<setprecision(3);
93      cout<<bsearch(1,2*n)<<endl;
94    }
95
96    return 0;
97  }
```

”

1.12 Cost_Flow

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (1000+10)
#define MAXM (10000*4+10)
#define MAXAi (35000)
#define eps (1e-3)
long long mul(long long a,long long b){return (a*b)%F;}
long long add(long long a,long long b){return (a+b)%F;}
long long sub(long long a,long long b){return
 ↪  (a-b+(a-b)/F*F+F)%F;}
typedef long long ll;

// PS double 替换成 int 不然会超时
//ps2: 注意 q 的大小
class Cost_Flow
{
public:
    int n,s,t;
    int q[10000000];
    int edge[MAXM],next[MAXM],pre[MAXN],weight[MAXM],size;
    double cost[MAXM];
    void addedge(int u,int v,int w,double c)
```

```
44      {
45          edge[++size]=v;
46          weight[size]=w;
47          cost[size]=c;
48          next[size]=pre[u];
49          pre[u]=size;
50      }
51      void addedge2(int u,int v,int w,double
        ↪  c){addedge(u,v,w,c),addedge(v,u,0,-c);}
52      bool b[MAXN];
53      double d[MAXN];
54      int pr[MAXN],ed[MAXN];
55      bool SPFA(int s,int t)
56      {
57          For(i,n) d[i]=INF;
58          MEM(b)
59          d[q[1]=s]=0;b[s]=1;
60          int head=1,tail=1;
61          while (head<=tail)
62          {
63              int now=q[head++];
64              Forp(now)
65              {
66                  int &v=edge[p];
67                  if (weight[p]&&d[now]+cost[p]<d[v])
68                  {
69                      d[v]=d[now]+cost[p];
70                      if (!b[v]) b[v]=1,q[++tail]=v;
71                      pr[v]=now,ed[v]=p;
72                  }
73              }
74              b[now]=0;
75          }
76          return fabs(d[t]-INF)>eps;
77      }
78      double totcost;
79
80      double CostFlow(int s,int t)
81      {
82          while (SPFA(s,t))
83          {
84              int flow=INF;
85              for(int x=t;x^s;x=pr[x]) flow=min(flow,weight[ed[x]]);
86              totcost+=(double)flow*d[t];
87              for(int x=t;x^s;x=pr[x])
                ↪  weight[ed[x]]-=flow,weight[ed[x]^1]+=flow;
```

```
88          }
89          return totcost;
90      }
91      void mem(int n,int t)
92      {
93          (*this).n=n;
94          size=1;
95          totcost=0;
96          MEM(pre) MEM(next)
97      }
98  }S;
99
100 int main()
101 {
102 //  freopen(".in","r",stdin);
103 //  freopen(".out","w",stdout);
104
105
106   return 0;
107 }
```

”

1.13 Cost_Flow_int

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (100000+10)
#define MAXM (100000+10)
long long mul(long long a,long long b){return (a*b)%F;}
long long add(long long a,long long b){return (a+b)%F;}
long long sub(long long a,long long b){return
 ↪  (a-b+(a-b)/F*F+F)%F;}
typedef long long ll;
class Cost_Flow
{
public:
    int n,s,t;
    int q[MAXM];
    int edge[MAXM],Next[MAXM],Pre[MAXN],weight[MAXM],size;
    int cost[MAXM];
    void addedge(int u,int v,int w,int c)
    {
        edge[++size]=v;
        weight[size]=w;
        cost[size]=c;
        Next[size]=Pre[u];
        Pre[u]=size;
```

```
44      }
45      void addedge2(int u,int v,int w,int
         ↪  c=0){addedge(u,v,w,c),addedge(v,u,0,-c);}
46      bool b[MAXN];
47      int d[MAXN];
48      int pr[MAXN],ed[MAXN];
49      bool SPFA(int s,int t)
50      {
51          For(i,n) d[i]=INF,b[i]=0;
52          d[q[1]=s]=0;b[s]=1;
53          int head=1,tail=1;
54          while (head<=tail)
55          {
56              int now=q[head++];
57              Forp(now)
58              {
59                  int &v=edge[p];
60                  if (weight[p]&&d[now]+cost[p]<d[v])
61                  {
62                      d[v]=d[now]+cost[p];
63                      if (!b[v]) b[v]=1,q[++tail]=v;
64                      pr[v]=now,ed[v]=p;
65                  }
66              }
67              b[now]=0;
68          }
69          return d[t]!=INF;
70      }
71      int totcost;
72
73      int CostFlow(int s,int t)
74      {
75        int maxflow=0;
76          while (SPFA(s,t))
77          {
78              int flow=INF;
79              for(int x=t;x^s;x=pr[x]) flow=min(flow,weight[ed[x]]);
80        totcost+=flow*d[t];
81        maxflow+=flow;
82              for(int x=t;x^s;x=pr[x])
                 ↪  weight[ed[x]]-=flow,weight[ed[x]^1]+=flow;
83          }
84  //        cout<<maxflow<<endl;
85          return totcost;
86      }
87      void mem(int n,int t)
```

```
    {
        (*this).n=n;
        size=1;
        totcost=0;
        MEM(Pre) MEM(Next)
    }
}S1;
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
int n,m;
int main()
{
//  freopen(".in","r",stdin);
//  freopen(".out","w",stdout);

  return 0;
}
```

”

1.14 Cost_Flow_upper_and_lower_bound

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (500+10)
#define MAXM ((100*3+10000)*12+10)
#define MAXAi (35000)
#define eps (1e-3)
long long mul(long long a,long long b){return (a*b)%F;}
long long add(long long a,long long b){return (a+b)%F;}
long long sub(long long a,long long b){return
 ↪  (a-b+(a-b)/F*F+F)%F;}
typedef long long ll;
class Cost_Flow
{
public:
    int n,s,t;
    int q[MAXM];
    int edge[MAXM],next[MAXM],pre[MAXN],weight[MAXM],size;
    int cost[MAXM];
    void addedge(int u,int v,int w,int c)
    {
        edge[++size]=v;
        weight[size]=w;
        cost[size]=c;
```

```
44          next[size]=pre[u];
45          pre[u]=size;
46      }
47      void addedge2(int u,int v,int w,int
        ↪  c){addedge(u,v,w,c),addedge(v,u,0,-c);}
48      bool b[MAXN];
49      int d[MAXN];
50      int pr[MAXN],ed[MAXN];
51      bool SPFA(int s,int t)
52      {
53          For(i,n) d[i]=INF,b[i]=0;
54          d[q[1]=s]=0;b[s]=1;
55          int head=1,tail=1;
56          while (head<=tail)
57          {
58              int now=q[head++];
59              Forp(now)
60              {
61                  int &v=edge[p];
62                  if (weight[p]&&d[now]+cost[p]<d[v])
63                  {
64                      d[v]=d[now]+cost[p];
65                      if (!b[v]) b[v]=1,q[++tail]=v;
66                      pr[v]=now,ed[v]=p;
67                  }
68              }
69              b[now]=0;
70          }
71          return d[t]!=INF;
72      }
73      int totcost;
74
75      int CostFlow(int s,int t)
76      {
77        int maxflow=0;
78          while (SPFA(s,t))
79          {
80              int flow=INF;
81              for(int x=t;x^s;x=pr[x]) flow=min(flow,weight[ed[x]]);
82        totcost+=flow*d[t];
83        maxflow+=flow;
84              for(int x=t;x^s;x=pr[x])
                ↪  weight[ed[x]]-=flow,weight[ed[x]^1]+=flow;
85          }
86  //        cout<<maxflow<<endl;
87          return totcost;
```

```
88      }
89      void mem(int n,int t)
90      {
91          (*this).n=n;
92          size=1;
93          totcost=0;
94          MEM(pre) MEM(next)
95      }
96  }S1;
97  int read()
98  {
99    int x=0,f=1; char ch=getchar();
100   while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
101   while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
102   return x*f;
103 }
104 int n,m;
105 int main()
106 {
107 //  freopen("bzoj2055.in","r",stdin);
108 //  freopen(".out","w",stdout);
109   n=read();m=read();
110   int s=2*n+1,t=s+1,S=t+1,T=S+1;
111   S1.mem(T,T);
112   const int inf = INF;
113   For(i,n) {
114     int v=read();
115     S1.addedge2(s,i,inf,0);
116     S1.addedge2(n+i,t,inf,0);
117
118     S1.addedge2(S,i+n,v,0);
119     S1.addedge2(i,T,v,0);
120   }
121   For(i,n) {
122     For(j,i+1,n) {
123       int c=read();
124       if (c==-1) continue;
125       S1.addedge2(i+n,j,inf,c);
126     }
127   }
128   S1.addedge2(t,s,m,0);
129
130   cout<<S1.CostFlow(S,T)<<endl;
131
132   return 0;
133 }
```

”

1.15 Dijkstra_heap

```cpp
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case %d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
typedef long long ll;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
struct Edge{
```

```
  int from,to,dist;
};
struct HeapNode {
  int d,u;
  bool operator< (const HeapNode& rhs) const {
    return d > rhs.d;
  }
};
#define MAXN (1000)
struct Dijkstra {
  int n,m;
  vector<Edge> edges;
  vector<int> G[MAXN];
  bool done[MAXN];
  int d[MAXN];
  int p[MAXN]; //最短路中上一条边
  int pnode[MAXN];
  void addedge(int u,int v,int w){
    edges.pb((Edge){u,v,w});
    G[u].pb(m++);
  }
  void addedge2(int u,int v,int w) {
    addedge(u,v,w);addedge(v,u,w);
  }
  void init(int _n){
    n=_n; m = 0;
    Rep(i,n) G[i].clear();
    edges.clear();
  }
  void dijkstra(int s) {
    priority_queue<HeapNode> Q;
    Rep(i,n) d[i]=INF,pnode[i]=-1;
    d[s]=0;
    MEM(done)
    Q.push((HeapNode){0,s});
    while(!Q.empty()) {
      HeapNode x=Q.top(); Q.pop();
      int u=x.u;
      if (done[u]) continue;
      done[u]=1;
      int mm=G[u].size();
      Rep(i,mm) {
        Edge e = edges[G[u][i]];
        if (d[e.to]>d[u]+e.dist) {
          d[e.to]=d[u]+e.dist;
          p[e.to]=G[u][i];
```

```
91              pnode[e.to]=u;
92              Q.push((HeapNode){d[e.to],e.to});
93            }
94          }
95        }
96      }
97  }S1;
98  int main()
99  {
100 //  freopen(".in","r",stdin);
101 //  freopen(".out","w",stdout);
102
103
104   return 0;
105 }
```

”

1.16 DLX

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
```

```
45
46  struct DLX {
47    #define maxn (10000000+10)
48    #define MAXNode (10000000+10)
49    #define maxr (10000000)
50    int n, sz;
51    int S[maxn];
52    int row[MAXNode],col[MAXNode];
53    int L[MAXNode],R[MAXNode],U[MAXNode],D[MAXNode];
54    int ansd, ans[maxr];
55
56    void init(int n) {
57      this->n = n;
58      Rep(i,n+1) {
59        U[i] = D[i] = i ; L[i]=i-1; R[i]=i+1;
60      }
61      R[n]=0; L[0]=n;
62      sz=n+1;
63      MEM(S)
64    }
65
66    void addRow(int r, vi columns) {
67      int fir = sz;
68      int cSz=columns.size();
69      Rep(i,cSz) {
70        int c=columns[i];
71        L[sz] = sz-1; R[sz] = sz+1; D[sz] = c; U[sz] = U[c];
72        D[ U[c] ] =sz; U[c]=sz;
73        row[sz] = r; col[sz] = c;
74        S[c]++; sz++;
75      }
76      R[sz-1]=fir; L[fir] = sz-1;
77    }
78    #define FOR(i,A,s) for(int i=A[s];i!=s;i=A[i])
79    void remove(int c) {
80      L[R[c]] = L[c];
81      R[L[c]] = R[c];
82      FOR(i,D,c)
83        FOR(j,R,i) {
84          U[D[j]] = U[j]; D[U[j]] = D[j]; --S[col[j]];
85        }
86    }
87    void restore(int c) {
88      FOR(i,U,c)
89        FOR(j,L,i) {
90          ++S[col[j]];
```

```
91          U[D[j]] = j;
92          D[U[j]] = j;
93        }
94      L[R[c]]=c;
95      R[L[c]]=c;
96    }
97    bool dfs(int d) {
98      if ( R[0] == 0) {
99        ansd = d;
100       return 1;
101     }
102     int c = R[0];
103     FOR(i,R,0) if (S[i] < S[c]) c = i;
104     remove(c);
105     FOR(i,D,c) {
106       ans[d] = row[i];
107       FOR(j,R,i) remove(col[j]);
108       if (dfs(d+1)) return 1;
109       FOR(j,L,i) restore(col[j]);
110     }
111     restore(c);
112     return 0;
113   }
114   bool solve(vi &v) {
115     v.clear();
116     if (!dfs(0)) return 0;
117     Rep(i,ansd) v.pb(ans[i]);
118     return 1;
119   }
120 };
121 DLX solver;
122 char puzzle[16][20];
123 bool init() {
124   Rep(i,16) {
125     if (scanf("%s",puzzle[i])!=1) return 0;
126   }
127   return 1;
128 }
129 const int SLOT = 0;
130 const int ROW = 1;
131 const int COL = 2;
132 const int SUB = 3;
133
134 int encode(int a,int b,int c) {
135   return a*16*16+b*16+c+1;
136 }
```

```
void decode(int code,int &a,int &b,int &c) {
  code--;
  c=code%16; code/=16;
  b=code%16; code/=16;
  a=code;
}
int main()
{
//  freopen("uva1309.in","r",stdin);
//  freopen(".out","w",stdout);
  bool fl=0;
  while(init()) {
    if (fl) puts("");  fl=1;
    solver.init(1024); //列数要初始化
    Rep(r,16) Rep(c,16) Rep(v,16) { // put v in (r,c)
      if (puzzle[r][c]=='-'||puzzle[r][c]=='A'+v) {
        vi columns;
        columns.pb(encode(SLOT,r,c));
        columns.pb(encode(ROW,r,v));
        columns.pb(encode(COL,c,v));
        columns.pb(encode(SUB,r/4*4+c/4,v));
        solver.addRow(encode(r,c,v),columns);
      }
    }
    vi ans;
    solver.solve(ans);
    int sz=SI(ans);
    Rep(i,sz) {
      int a,b,c;
      decode(ans[i],a,b,c);
      puzzle[a][b]=c+'A';
    }
    Rep(r,16) puts(puzzle[r]);
  }


  return 0;
}
```

”

1.17 DLX_exact_extra

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (100000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case %d: %d\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
int ans=0;
```

```
45  struct DLX {
46    #define MAXNode (240000+10)
47    #define maxn (600+10)
48    #define maxm (600+10)
49    int L[MAXNode],R[MAXNode],U[MAXNode],D[MAXNode];
50    int sz,col[MAXNode],S[maxm],H[maxn];
51    int n;
52    bool vis[maxm];
53    void init(int m) {
54      Rep(i,m+1) {
55        L[i] = i-1;
56        R[i] = i+1;
57        U[i] = D[i] =i;
58        S[i] = 0;
59      }
60      memset(H,-1,sizeof(H));
61      L[0]=m; R[m] = 0;
62      sz=m+1;
63    }
64    #define FOR(i,A,s) for(int i=A[s];i!=s;i=A[i])
65    void remove(int c) {
66      FOR(i,D,c) {
67        L[R[i]] = L[i];
68        R[L[i]] = R[i];
69      }
70    }
71    void resume(int c) {
72      int i;
73      for(int i=U[c];i!=c;i=U[i]) {
74        L[R[i]] = R[L[i]] = i;
75      }
76    }
77
78    //精确覆盖
79    void remove1(int c) {
80      L[R[c]] = L[c];
81      R[L[c]] = R[c];
82      FOR(i,D,c)
83        FOR(j,R,i) {
84          U[D[j]] = U[j]; D[U[j]] = D[j]; --S[col[j]];
85        }
86    }
87    void resume1(int c) {
88      FOR(i,U,c)
89        FOR(j,L,i) {
```

```
90          ++S[col[j]];
91          U[D[j]] = j;
92          D[U[j]] = j;
93        }
94      L[R[c]]=c;
95      R[L[c]]=c;
96    }
97
98    void link(int r,int c) {
99      U[sz] = c;
100     D[sz] = D[c];
101     U[D[c]] = sz;
102     D[c] = sz;
103     if (H[r]==-1) { H[r]=L[sz]=R[sz]=sz;  }
104     else {
105       L[sz] =H[r];
106       R[sz] = R[H[r]];
107       L[R[H[r]]] = sz;
108       R[H[r]] = sz;
109     }
110     S[c]++ ; col[sz++]=c;
111   }
112   int A() {
113     MEM(vis)
114     int res=0;
115     FOR(i,R,0) {
116       if(i>n) break;
117       if (!vis[i]) {
118         res++;
119         vis[i] = 1;
120         FOR(j,D,i) FOR(k,R,j) {
121           vis[col[k]]=1;
122         }
123       }
124     }
125     return res;
126   }
127   // exact 1..n extra >n
128   void dfs(int d) {
129     if (!R[0]||R[0]>n)  {
130       ans=min(ans,d);
131     } else if (d+A()<ans) {
132       int c=R[0];
133       FOR(i,R,0) {
134         if(i>n) break;
135         if (S[i]<S[c]) c=i;
```

```
136         }
137         FOR(i,D,c) {
138           remove(i);
139           FOR(j,R,i) if (col[j]<=n) remove(j);
140           FOR(j,R,i) if (col[j]>n) remove1(col[j]);
141 
142           dfs(d+1);
143           FOR(j,L,i) if (col[j]>n) resume1(col[j]);
144           FOR(j,L,i) if (col[j]<=n) resume(j);
145           resume(i);
146         }
147       }
148     }
149 }solver;
150 int n,m;
151 #define MAXN (30)
152 int main()
153 {
154 //  freopen("hdu3957.in","r",stdin);
155 //  freopen(".out","w",stdout);
156   int T=read();
157   For(kcase,T) {
158     cin>>n;
159     solver.init(3*n);
160     solver.n=2*n;
161     int t=0;
162     Rep(i,n) {
163       cin>>m;
164       Rep(j,m) {
165         int u=2*i+j+1;
166         int k=read();
167         Rep(l,m) solver.link(u,2*i+l+1);
168         while(k--) {
169           int a=read(),b=read();
170           if (a==i) continue;
171           int v=a*2+b+1;
172           solver.link(u,v);
173         }
174         solver.link(u,2*n+i+1);
175       }
176       if (m==1) solver.link(2*i+2,2*i+2),++t;
177     }
178     ans=INF;
179     solver.dfs(0);
180     Pr(kcase,ans-t);
181   }
```

```
182
183     return 0;
184 }
```

”

1.18 DLX_extra

```
#pragma comment(linker, "/STACK:102400000,102400000")
#include <stdio.h>
#include <iostream>
#include <algorithm>
#include <sstream>
#include <stdlib.h>
#include <string.h>
#include <limits.h>
#include <string>
#include <time.h>
#include <math.h>
#include <queue>
#include <stack>
#include <set>
#include <map>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (100000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
```

```cpp
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
int ans=0;
struct DLX {
  #define MAXNode (1000000+10)
  #define maxn (1000000+10)
  #define maxm (1000)
  int L[MAXNode],R[MAXNode],U[MAXNode],D[MAXNode];
  int sz,col[MAXNode],S[maxm],H[maxn];
  bool vis[maxm];
  void init(int m) {
    Rep(i,m+1) {
      L[i] = i-1;
      R[i] = i+1;
      U[i] = D[i] =i;
      S[i] = 0;
    }
    memset(H,-1,sizeof(H));
    L[0]=m; R[m] = 0;
    sz=m+1;
  }
  #define FOR(i,A,s) for(int i=A[s];i!=s;i=A[i])
  void remove(int c) {
    FOR(i,D,c) {
      L[R[i]] = L[i];
      R[L[i]] = R[i];
    }
  }
  void resume(int c) {
    int i;
    for(int i=U[c];i!=c;i=U[i]) {
      L[R[i]] = R[L[i]] = i;
    }
  }
```

```
90    void link(int r,int c) {
91      U[sz] = c;
92      D[sz] = D[c];
93      U[D[c]] = sz;
94      D[c] = sz;
95      if (H[r]==-1) { H[r]=L[sz]=R[sz]=sz;  }
96      else {
97        L[sz] =H[r];
98        R[sz] = R[H[r]];
99        L[R[H[r]]] = sz;
100       R[H[r]] = sz;
101     }
102     S[c]++ ; col[sz++]=c;
103   }
104   int A() {
105     MEM(vis)
106     int res=0;
107     FOR(i,R,0) {
108       if (!vis[i]) {
109         res++;
110         vis[i] = 1;
111         FOR(j,D,i) FOR(k,R,j) {
112           vis[col[k]]=1;
113         }
114       }
115     }
116     return res;
117   }
118   void dfs(int d) {
119     if (!R[0])  {
120       ans=min(ans,d);
121     } else if (d+A()<ans) {
122       int c=R[0];
123       FOR(i,R,0) {
124         if (S[i]<S[c]) c=i;
125       }
126       FOR(i,D,c) {
127         remove(i);
128         FOR(j,R,i) remove(j);
129         dfs(d+1);
130         FOR(j,L,i) resume(j);
131         resume(i);
132       }
133     }
134    }
135 }solver;
```

```
136 int main()
137 {
138 //  freopen(".in","r",stdin);
139 //  freopen(".out","w",stdout);
140
141
142   return 0;
143 }
```

”

1.19 dynamicsSegmentTree_Treedivided

```
#include<cstdio>
#include<cstring>
#include<algorithm>
#include<functional>
#include<cctype>
#include<cstdlib>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,0x3f,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEMx(a,b) memset(a,b,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
```

```
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
#define Lson (ls[o])
#define Rson (rs[o])
int ls[10000000],rs[10000000],sum[10000000],mx[10000000],size;
void pushUp(int o) {
  sum[o]=sum[Lson]+sum[Rson];
  mx[o]=max(mx[Lson],mx[Rson]);
}
void mem(){size=0;}
void update(int l,int r,int &o,int x,int c) {
  if (!o) {
    o=++size;
    ls[o]=rs[o]=sum[o]=mx[o]=0;
  }
  if (l==r) {
    sum[o]=mx[o]=c; return;
  }
  int m=(l+r)>>1;
  if (x<=m) update(l,m,Lson,x,c);
  else update(m+1,r,Rson,x,c);
  pushUp(o);
}
int querySum(int l,int r,int o,int L,int R) {
  if (!o) return 0;
  if (L<=l&&r<=R) return sum[o];
  int m=(l+r)>>1;
  int ret=0;
  if (L<=m) ret+=querySum(l,m,Lson,L,R);
  if (m<R) ret+=querySum(m+1,r,Rson,L,R);
  return ret;
}
int queryMx(int l,int r,int o,int L,int R) {
  if (!o) return 0;
  if (L<=l&&r<=R) return mx[o];
  int m=(l+r)>>1;
  int ret=0;
  if (L<=m) ret = max(ret, queryMx(l,m,Lson,L,R));
  if (m<R) ret = max(ret, queryMx(m+1,r,Rson,L,R));
  return ret;
}

#define MAXN (200000+10)
```

```cpp
int n,m;
int w[MAXN],c[MAXN],root[MAXN];
struct Tree{
  #define MAXM (200000+10)
  void mem(){MEM(Pre) siz=1;}
  int edge[MAXM],Next[MAXM],Pre[MAXN],siz;
  void addedge(int u,int v)
  {
    edge[++siz]=v;
    Next[siz]=Pre[u];
    Pre[u]=siz;
  }
  void addedge2(int u,int v){addedge(u,v);addedge(v,u);}
  bool vis[MAXN];
  int cnt,id[MAXN];
  int son[MAXN],dep[MAXN],sz[MAXN],top[MAXN],pre[MAXN],q[MAXN];
  void build()
  {
    MEM(vis) cnt=0; MEM(id)
    MEM(son) MEM(dep) MEM(sz) MEM(top) MEM(pre) MEM(q)
    int r=1;
    vis[dep[1]=q[1]=1]=1;
    For(i,r)
    {
      int u=q[i];
      Forp(u)
      {
        int v=edge[p];
        if (vis[v]) continue; else vis[v]=1;
        dep[ q[++r]=v ]=dep[u]+1;
        pre[v]=u;
      }
    }
    ForD(i,r) {
      sz[pre[q[i]]] += ++sz[q[i]];
      if (sz[son[pre[q[i]]]]<sz[q[i]] ) son[pre[q[i]]] = q[i];
    }
    For(i,r) {
      if (!top[q[i]])
        for(int x=q[i];x;x=son[x]) {
          top[x]=q[i];
          id[x]=++cnt;
        }
    }

  }
```

```
137     int lca(int a,int b)
138     {
139       while(1) {
140         if (top[a]==top[b]) return dep[a]<=dep[b] ? a:b;
141         if (dep[top[a]]<dep[top[b]]) swap(a,b);
142         a=pre[top[a]];
143       }
144     }
145
146     ll AskSum(int p,int a,int b)
147     {
148       ll ans=0;
149       while (top[a]^top[b]) {
150         ans+=querySum(1,n,root[p],id[top[a]],id[a]);
151         a=pre[top[a]];
152       }
153       ans+=querySum(1,n,root[p],id[b],id[a]);
154       return ans;
155     }
156     ll AskMx(int p,int a,int b)
157     {
158       int ans=0;
159       while (top[a]^top[b]) {
160         ans=max(ans,queryMx(1,n,root[p],id[top[a]],id[a]));
161         a=pre[top[a]];
162       }
163       ans=max(ans,queryMx(1,n,root[p],id[b],id[a]));
164       return ans;
165     }
166     void set(int &o,int a,int c)
167     {
168       update(1,n,o,id[a],c);;
169     }
170   }S;
171   int main()
172   {
173   //  freopen("bzoj3531.in","r",stdin);
174   //  freopen("bzoj3531.out","w",stdout);
175     scanf("%d%d",&n,&m);
176     MEM(root)
177     For(i,n) {
178       scanf("%d%d",&w[i],&c[i]);
179     }
180     mem();
181     S.mem();
182     For(i,n-1) S.addedge2(read(),read());
```

```
183   S.build();
184   For(i,n)
185     S.set(root[c[i]],i,w[i]);
186
187
188   while(m--) {
189     char op[10]; int a,b;
190     scanf("%s%d%d",op,&a,&b);
191     int t=S.lca(a,b);
192     if (strcmp(op,"QS")==0) {
193        printf("%lld\n",S.AskSum(c[b],a,t)+S.AskSum(c[b],b,t)-
           ↪  S.AskSum(c[b],t,t));
194     } else if (strcmp(op,"QM")==0){
195        printf("%lld\n",max(S.AskMx(c[b],a,t),S.AskMx(c[b],b,t)));
196     } else if (strcmp(op,"CC")==0) {
197       S.set(root[c[a]],a,0);
198       c[a]=b;
199       S.set(root[c[a]],a,w[a]);
200     } else {
201       w[a]=b;
202       S.set(root[c[a]],a,w[a]);
203     }
204   }
205   return 0;
206 }
```

”

1.20 gauss_xor

```cpp
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (10086)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
typedef long long ll;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
#define MAXN (100000+10)
int n;
ll a[MAXN];
void gauss(int n) {
  For(i,n) {
    Fork(j,i+1,n) if (a[j]>a[i]) swap(a[i],a[j]);
    if (!a[i]) return ;
    For(k,n) if (i^k) a[k]=min(a[k],a[k]^a[i]);
  }
}
int main()
```

```
45 {
46 //  freopen(".in","r",stdin);
47 //  freopen(".out","w",stdout);
48   n=read();
49   For(i,n) a[i]=read();
50   gauss(n);
51   return 0;
52 }
```

”

1.21 geometry3D

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
```

```
}
ll sqr(ll a){return a*a;}
ld sqr(ld a){return a*a;}
double sqr(double a){return a*a;}

ld PI = 3.14159265358979323846264338328;
const double eps=1e-10;
int dcmp(double x) {
  if (fabs(x)<eps) return 0; else return x<0 ? -1 : 1;
}
struct P3{
  double x,y,z;
  P3(double x=0,double y=0,double z=0):x(x),y(y),z(z){}
};
typedef P3 V3;
V3 operator+(V3 A,V3 B) {
  return V3(A.x+B.x, A.y+B.y ,A.z+B.z);
}
V3 operator-(V3 A,V3 B) {
  return V3(A.x-B.x, A.y-B.y ,A.z-B.z );
}
V3 operator*(V3 A,double p) {
  return V3(A.x*p, A.y*p, A.z*p);
}
V3 operator/(V3 A,double p) {
  return V3(A.x/p, A.y/p, A.z/p);
}
double Dot(V3 A,V3 B) {return A.x*B.x + A.y*B.y + A.z*B.z; }
double Length(V3 A) {return sqrt(Dot(A,A));}
double Angle(V3 A,V3 B){return acos(Dot(A, B)) / Length(A) /
 ↪  Length(B); }
bool operator==(const P3& a,const P3& b) {
  return dcmp(a.x-b.x)==0 && dcmp(a.y-b.y) == 0 && dcmp(a.z-b.z)
   ↪  == 0;
}
double DistanceToPlane(const P3& p,const P3& p0,const V3& n) {
  return fabs(Dot(p-p0,n));
}
P3 GetPlaneProjection(const P3& p,const P3 &p0, const V3 &n) {
  return p-n*Dot(p-p0, n);
}
P3 LinePlaneProjection(P3 p1,P3 p2,P3 p0,V3 n) {
  V3 v=p2-p1;
  double t = (Dot(n,p0-p1) / Dot(n,p2-p1));
  return p1+v*t;
}
```

```
V3 Cross(V3 A,V3 B) {
  return V3(A.y*B.z - A.z*B.y , A.z*B.x - A.x * B.z, A.x*B.y -
   ↪  A.y*B.x );
}
double Area2(P3 A,P3 B,P3 C) {return Length(Cross(B-A,C-A));}
bool PointInTri(P3 p,P3 p0,P3 p1,P3 p2) {
  double area1 = Area2(p,p0,p1);
  double area2 = Area2(p,p1,p2);
  double area3 = Area2(p,p2,p0);
  return dcmp(area1+area2+area3 - Area2(p0, p1, p2) ) == 0;
}
bool TriSegIntersection(P3 p0, P3 p1 ,P3 p2, P3 A, P3 B, P3 &p) {
  V3 n = Cross(p1-p0,p2-p0);
  if (dcmp(Dot(n, B-A) ) ==0 ) return 0; //平行，共面
  else {
    double t = Dot(n,p0-A) / Dot(n, B-A );
    if (dcmp(t)<0 || dcmp(t-1)>0 ) return 0;
    p = A + (B-A) * t;
    return PointInTri(p,p0,p1,p2);
  }
}
double DistanceToLine(P3 p,P3 A,P3 B) {
  V3 v1= B-A , v2 = p-A;
  return Length(Cross(v1,v2))/Length(v1);
}
double DistanceToSegment(P3 p,P3 A,P3 B) {
  if (A==B) return Length(p-A);
  V3 v1 = B - A , v2 = p - A , v3 = p - B ;
  if (dcmp(Dot(v1, v2) < 0)) return Length(v2);
  else if (dcmp(Dot(v1, v3)) > 0) return Length(v3);
  else return Length(Cross(v1,v2)) / Length(v1);
}
double Volume6(P3 A, P3 B, P3 C, P3 D) {
  return Dot(D-A, Cross(B-A ,C-A));
}
struct Face{
  int v[3];
  V3 normal(P3 *p) const {
    return Cross(p[v[1]]-p[v[0]], p[v[2]]-p[v[0]]);
  }
  int cansee(P3 *p, int i) const {
    return Dot(p[i] - p[v[0]], normal(p))>0? 1 : 0;
  }
};
bool vis[1000][1000];
vector<Face> CH3D(P3 *p, int n) {
```

```
134   MEM(vis);
135   vector<Face> cur;
136   cur.pb((Face){{0,1,2}});
137   cur.pb((Face){{2,1,0}});
138   Fork(i,3,n-1) {
139     vector<Face> next;
140     int sz=SI(cur);
141     Rep(j,sz) {
142       Face &f = cur[j];
143       int res = f.cansee(p, i);
144       if (!res) next.pb(f);
145       Rep(k,3) vis[f.v[k]][f.v[(k+1)%3]] = res;
146     }
147     Rep(j,sz)
148       Rep(k,3) {
149         int a=cur[j].v[k], b=cur[j].v[(k+1)%3];
150         if (vis[a][b]!= vis[b][a] && vis[a][b]) {
151           next.pb((Face) {a, b, i});
152         }
153       }
154     cur = next;
155   }
156   return cur;
157 }
158 double rand01() {return rand()/(double)RAND_MAX;}
159 double randeps() {return (rand01()-0.5)*eps; }
160 P3 add_noise(P3 p) {
161   return P3(p.x+randeps(),p.y+randeps(),p.z+randeps());
162 }
163 bool TriTriIntersection(P3 *T1, P3 *T2) {
164   P3 p;
165   Rep(i,3) {
166     if (TriSegIntersection(T1[0],T1[1],T1[2],T2[i],T2[(i+1)%3],p))
      ↪  return 1;
167     if (TriSegIntersection(T2[0],T2[1],T2[2],T1[i],T1[(i+1)%3],p))
      ↪  return 1;
168     return 0;
169   }
170 }
171 P3 read_point3() {
172   P3 a;
173   scanf("%lf%lf%lf",&a.x,&a.y,&a.z);
174   return a;
175 }
176 bool LineDistance3D(P3 p1, V3 u, P3 p2, V3 v, double &s) {
177   double b = Dot(u,u)*Dot(v,v) - Dot(u,v) * Dot(u,v);
```

```
  if (dcmp(b) == 0 ) return 0;
  double a = Dot(u,v)*Dot(v,p1-p2) - Dot(v,v) * Dot(u,p1-p2);
  s=a/b;
  return 1;
}

//prism centroid
P3 Centroid(P3 A,P3 B,P3 C,P3 D){
  return (A+B+C+D)/4.0;
}
//3d-convex hulls centroid
P3 Centroid(vector<Face>& v,P3* p) {
  int n=v.size();
  P3 C=v[0].v[0], tot;
  double totv=0;
  Rep(i,n) {
    P3 p0=p[v[i].v[0]],p1=p[v[i].v[1]],p2=p[v[i].v[2]];
    double v = -(Volume6(p0,p1,p2,C));
    totv+=v;
    tot = tot + Centroid(p0,p1,p2,C)*v;
  }
  P3 C2=tot/totv;
  return C2;
}
int main()
{
//  freopen(".in","r",stdin);
//  freopen(".out","w",stdout);



  return 0;
}
```

”

1.22 Gusfield

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<vector>
#include<iostream>
#include<cmath>
#include<set>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define SI(x) ((x).size())
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define vi vector<int>
#define pb push_back
#define MAXN (200+100)
#define MAXM (40000*2+100)
typedef long long ll;
long long mul(long long a,long long b){return (a*b)%F;}
long long add(long long a,long long b){return (a+b)%F;}
long long sub(long long a,long long b){return
 ↪  (a-b+(a-b)/F*F+F)%F;}
int gmin(int &a,int b) {return a=min(a,b);}
int read()
{
    int x=0,f=1; char ch=getchar();
    while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
    while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
    return x*f;
}
```

```
class Max_flow  //dinic+???????????
{
public:
    int n,t;
    int q[MAXN];
    int edge[MAXM],Next[MAXM],Pre[MAXN],weight[MAXM],size;
    void addedge(int u,int v,int w)
    {
        edge[++size]=v;
        weight[size]=w;
        Next[size]=Pre[u];
        Pre[u]=size;
    }
    void addedge2(int u,int v,int
    ↪  w){addedge(u,v,w),addedge(v,u,0);}
    bool b[MAXN];
    int d[MAXN];
    bool SPFA(int s,int t)
    {
        For(i,n) d[i]=INF;
        MEM(b)
        d[q[1]=s]=0;b[s]=1;
        int head=1,tail=1;
        while (head<=tail)
        {
            int now=q[head++];
            Forp(now)
            {
                int &v=edge[p];
                if (weight[p]&&!b[v])
                {
                    d[v]=d[now]+1;
                    b[v]=1,q[++tail]=v;
                }
            }
        }
        return b[t];
    }
    int iter[MAXN];
    int dfs(int x,int f)
    {
        if (x==t) return f;
        Forpiter(x)
        {
            int v=edge[p];
            if (weight[p]&&d[x]<d[v])
```

```
            {
                int nowflow=dfs(v,min(weight[p],f));
                if (nowflow)
                {
                  weight[p]-=nowflow;
                  weight[p^1]+=nowflow;
                  return nowflow;
                }
            }
        }
        return 0;
    }
    int max_flow(int s,int t)
    {
        (*this).t=t;
        int flow=0;
        while(SPFA(s,t))
        {
            For(i,n) iter[i]=Pre[i];
            int f;
            while (f=dfs(s,INF))
                flow+=f;
        }
        return flow;
    }
    void mem(int n)
    {
        (*this).n=n;
        size=1;
        For(i,n) Pre[i]=0;
    }
}S;
int n,m,f[MAXN];
int g[MAXN][MAXN];
int ans[MAXN][MAXN];
int cut(int u,int v){
    S.mem(n);
    For(i,n) For(j,n) if (i!=j){
        S.addedge2(i,j,g[i][j]);
    }
    return S.max_flow(u,v);
}
int main()
{
//  freopen("uva11594.in","r",stdin);
//  freopen(".out","w",stdout);
```

```
    int T=read();
    For(tcase,T) {
        printf("Case #%d:\n",tcase);
        n=read();
        MEMI(ans) For(i,n) ans[i][i]=0;
        For(i,n) For(j,n) g[i][j]=read();

        For(i,n) f[i]=1;
        Fork(i,2,n) {
            int v=f[i];
            int p=cut(i,v);
            vi v1,v2;
            For(j,n) if (1) {
                if (S.b[j]) v1.pb(j);
                else v2.pb(j);
            }
            // Rep(i,SI(v1)) cout<<v1[i]<<' ';cout<<endl;
            // Rep(j,SI(v2)) cout<<v2[j]<<' ';cout<<endl;

            Rep(i,SI(v1)) Rep(j,SI(v2)) {
                gmin(ans[v1[i]][v2[j]],p);
                gmin(ans[v2[j]][v1[i]],p);

            }
            // For(j,i)
             ↪  gmin(ans[i][j],min(p,ans[f[i]][j])),gmin(ans[j][i],min(p,ans[f[i]][j]));
            Fork(j,i,n) {
                if (f[j]==v&&S.b[j]) f[j]=i;
            }
        }
        For(i,n) {
            For(j,n-1) printf("%d ",ans[i][j]);
            printf("%d\n",ans[i][n]);
        }

    }
    return 0;
}
```

”

1.23 hanshushi_Segmenttree

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
#include<iomanip>
#include<vector>
#include<string>
#include<queue>
#include<stack>
#include<map>
#include<sstream>
#include<ext/rope>
using namespace std;
using namespace __gnu_cxx;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %I64d\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
```

```
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
#define MAXN (200000+10)
int n,m,a[MAXN],a2[MAXN];
struct node
{
  node *ch[2];
  int a,siz;
  node(){ch[0]=ch[1]=NULL;siz=a=0;}
  void update()
  {
    siz=a;
    if (ch[0]) siz+=ch[0]->siz;
    if (ch[1]) siz+=ch[1]->siz;
  }
}*null=new node(),*root[MAXN]={NULL},q[MAXN*9];
int q_s;
void make_node(node *&y,node *&x,int l,int r,int t)
{
  if (x==NULL) x=null;
  y=&q[++q_s];
  *y=node();
  int m=(l+r)>>1;
  if (l==r)
  {
    *y=*x;
    y->siz++;y->a++;
    return;
  }
  if (t<=a2[m])
```

```
91    {
92      make_node(y->ch[0],x->ch[0],l,m,t);
93      y->ch[1]=x->ch[1];
94      y->update();
95    }
96    else
97    {
98      make_node(y->ch[1],x->ch[1],m+1,r,t);
99      y->ch[0]=x->ch[0];
100     y->update();
101   }
102 }
103 void find(node *&x1,node *&x2,int l,int r,int k)
104 {
105   if (x1==NULL) x1=null;
106   if (x2==NULL) x2=null;
107   if (l==r) {printf("%d\n",a2[l]);return;}
108   int m=(l+r)>>1;
109   int ls=0;
110   if (x2->ch[0]) ls+=x2->ch[0]->siz;
111   if (x1->ch[0]) ls-=x1->ch[0]->siz;
112   if (ls>=k) find(x1->ch[0],x2->ch[0],l,m,k);
113   else find(x1->ch[1],x2->ch[1],m+1,r,k-ls);
114 }
115
116 int main()
117 {
118 //  freopen(".in","r",stdin);
119 //  freopen(".out","w",stdout);
120   null->ch[0]=null; null->ch[1]=null;
121   q_s=0;
122   scanf("%d%d",&n,&m);
123   For(i,n) scanf("%d",&a[i]),a2[i]=a[i];
124   sort(a2+1,a2+1+n);
125   int size=unique(a2+1,a2+1+n)-(a2+1);
126   For(i,n)
127   {
128     make_node(root[i],root[i-1],1,size,a[i]);
129   }
130   For(i,m)
131   {
132     int l,r,k;
133     scanf("%d%d%d",&l,&r,&k);
134     find(root[l-1],root[r],1,size,k);
135   }
136
```

```
137
138
139     return 0;
140 }
```

”

1.24 Int128

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
#include<iomanip>
#include<vector>
#include<string>
#include<queue>
#include<stack>
#include<map>
#include<sstream>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %I64d\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
```

```
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}

struct Int_128{
    ull a,b;
    Int_128(ll x){a=0,b=x;}
    friend bool operator < (Int_128 x,Int_128 y)
    {
        return x.a<y.a||x.a==y.a&&x.b<y.b;
    }
    friend Int_128 operator + (Int_128 x,Int_128 y)
    {
        Int_128 re(0);
        re.a=x.a+y.a+(x.b+y.b<x.b);
        re.b=x.b+y.b;
        return re;
    }
    friend Int_128 operator - (Int_128 x,Int_128 y)
    {
        y.a=~y.a;y.b=~y.b;
        return x+y+1;
    }
    void Div2()
    {
        b>>=1;b|=(a&1ll)<<63;a>>=1;
    }
    friend Int_128 operator * (Int_128 x,Int_128 y)
    {
        Int_128 re=0;
        while(y.a||y.b)
        {
            if(y.b&1)re=re+x;
```

```
            x=x+x;y.Div2();
        }
        return re;
    }
    friend Int_128 operator % (Int_128 x,Int_128 y)
    {
        Int_128 temp=y;
        int cnt=0;
        while(temp<x)temp=temp+temp,++cnt;
        for(;cnt>=0;cnt--)
        {
            if(temp<x)x=x-temp;
            temp.Div2();
        }
        return x;
    }
};

int main()
{
  freopen("A.in","r",stdin);
//  freopen(".out","w",stdout);
  int T=read();
  while(T--) {

  }


  return 0;
}
```

”

1.25 KD_Tree

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (1000000000)
#define F (100000007)
#define MAXN (500000+10)
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int n;

int cmp_d=0;
class node
{
public:
  int x[2];
  int l,r,minv[2],maxv[2];

  node(){}
  node(int a,int b){MEM(x) l=r=0; x[0]=a,x[1]=b; Rep(i,2)
  ↪  minv[i]=maxv[i]=x[i];}


```

```
44
45    int& operator[](int i){return x[i];  }
46  };
47
48  int dis(node a,node b){
49    int ans=0;
50    Rep(i,2) ans+=abs(a.x[i]-b.x[i]);
51    return ans;
52  }
53
54  int dis2(node p,node a) // 点 p 和方形区域 a 的欧几里德距离
55  {
56    int ans=0;
57    Rep(i,2)
58    {
59      if (p.x[i]<a.minv[i]) ans+=a.minv[i]-p.x[i];
60      else
61      if (p.x[i]>a.maxv[i]) ans+=p.x[i]-a.maxv[i];
62    }
63    return ans;
64  }
65
66
67  int cmp(node a,node b){return a[cmp_d]<b[cmp_d];  }
68
69  class KD_Tree
70  {
71  public:
72    node a[MAXN*3];
73    KD_Tree()
74    {
75    }
76
77    void mem()
78    {
79    }
80
81    void update(node& o)
82    {
83      if (o.l)
84      {
85        node p=a[o.l];
86        Rep(i,2) o.minv[i]=min(o.minv[i],p.minv[i]);
87        Rep(i,2) o.maxv[i]=max(o.maxv[i],p.maxv[i]);
88      }
89      if (o.r)
```

```
90      {
91        node p=a[o.r];
92        Rep(i,2) o.minv[i]=min(o.minv[i],p.minv[i]);
93        Rep(i,2) o.maxv[i]=max(o.maxv[i],p.maxv[i]);
94      }
95
96    }
97
98    int build(int L,int R,int nowd)
99    {
100     int m=(L+R)>>1;
101
102     cmp_d=nowd;
103     nth_element(a+L+1,a+m+1,a+R+1,cmp);
104
105     if (L^m) a[m].l=build(L,m-1,nowd^1);
106     if (R^m) a[m].r=build(m+1,R,nowd^1);
107
108     update(a[m]);
109
110     return m;
111
112   }
113
114   int root;
115   void _build(int L,int R,int nowd) //1-n 的节点 至少为 1
116   {
117     root=build(L,R,nowd);
118   }
119
120   void insert(int o,int k,int nowd)
121   {
122     int p=a[o].x[nowd];
123     int p2=a[k].x[nowd];
124
125     if (p2<=p)
126     {
127       if (a[o].l)
128         insert(a[o].l,k,nowd^1);
129       else a[o].l=k;
130     }
131     else
132     {
133       if (a[o].r)
134         insert(a[o].r,k,nowd^1);
135       else a[o].r=k;
```

```
136
137       }
138
139
140       update(a[o]);
141
142     }
143     void _insert(int k,int nowd)
144     {
145       int p=root;
146       insert(root,k,nowd);
147     }
148
149
150     node _p;
151     int _ans;
152
153     void ask_min_dis(int o)
154     {
155       if (o==0) return;
156       _ans=min(_ans,dis(a[o],_p));
157
158       int ans1=a[o].l ? dis2(_p,a[a[o].l]) : INF;  // 点 p 到区域内任
          ↪ 意一点的距离的最小值
159       int ans2=a[o].r ? dis2(_p,a[a[o].r]) : INF;
160
161
162
163       if (ans1<ans2)
164       {
165         if(ans1<_ans) ask_min_dis(a[o].l);
166         if(ans2<_ans) ask_min_dis(a[o].r);
167       }
168       else {
169         if(ans2<_ans) ask_min_dis(a[o].r);
170         if(ans1<_ans) ask_min_dis(a[o].l);
171       }
172
173
174     }
175
176     int _ask(node p)
177     {
178       _p=p;_ans=INF;
179       ask_min_dis(root);
```

```
180       return _ans;
181     }
182
183   }S;
184   int main()
185   {
186     For(i,n)
187     {
188       int x,y;
189       scanf("%d%d",&x,&y);
190       S.a[i]=node(x,y);
191     }
192     S.a[++n]=node(INF,INF);
193     S._build(1,n,0);
194     return 0;
195   }
```

”

1.26 KD_Tree2

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (1000000000)
#define F (100000007)
#define MAXN (200000+10)
#define fac 0.65
typedef long long ll;
int n;

int cmp_d=0;
class node
{
public:
  int x[2];
  int l,r,minv[2],maxv[2];
  int w,sumv,siz;
  node(){}
  node(int a,int b,int _w){l=r=0; siz=1; w=sumv=_w; x[0]=a,x[1]=b;
  ↪  Rep(i,2) minv[i]=maxv[i]=x[i];}
  int& operator[](int i){return x[i];  }
};

int cmp(node a,node b){return a[cmp_d]<b[cmp_d];  }

```

```
44
45  int cmp2(int i,int j);
46
47  int p;
48  char c;
49  int read()
50  {
51    while (c=getchar(),!isdigit(c));
52    p=c-'0';
53    while (isdigit(c=getchar())) p=p*10+c-'0'; return p;
54  }
55  class KD_Tree
56  {
57  public:
58    node a[MAXN];
59
60    void update(node& o)
61    {
62      o.sumv=o.w;
63      o.minv[0]=o.maxv[0]=o.x[0];o.minv[1]=o.maxv[1]=o.x[1];
64      o.siz=1;
65      if (o.l)
66      {
67        node p=a[o.l];
68        Rep(i,2) o.minv[i]=min(o.minv[i],p.minv[i]);
69        Rep(i,2) o.maxv[i]=max(o.maxv[i],p.maxv[i]);
70        o.sumv+=p.sumv;
71        o.siz+=p.siz;
72      }
73      if (o.r)
74      {
75        node p=a[o.r];
76        Rep(i,2) o.minv[i]=min(o.minv[i],p.minv[i]);
77        Rep(i,2) o.maxv[i]=max(o.maxv[i],p.maxv[i]);
78        o.sumv+=p.sumv;
79        o.siz+=p.siz;
80      }
81
82    }
83
84    int build(int L,int R,int nowd,node *a)
85    {
86      int m=(L+R)>>1;
87
88      cmp_d=nowd;
89      nth_element(a+L+1,a+m+1,a+R+1,cmp);
```

```
90
91      if (L^m) a[m].l=build(L,m-1,nowd^1,a);
92      if (R^m) a[m].r=build(m+1,R,nowd^1,a);
93
94      update(a[m]);
95
96      return m;
97
98    }
99
100   int po[MAXN],pt;
101   void dfs(int x)
102   {
103     po[++pt]=x;
104     if (a[x].l) dfs(a[x].l);
105     if (a[x].r) dfs(a[x].r);
106   }
107
108   int rebuild(int L,int R,int nowd)
109   {
110     int m=(L+R)>>1;
111
112     cmp_d=nowd;
113     nth_element(po+L+1,po+m+1,po+R+1,cmp2);
114     int now=po[m];
115     a[now].l=a[now].r=0;
116     if (L^m) a[now].l=rebuild(L,m-1,nowd^1);
117     if (R^m) a[now].r=rebuild(m+1,R,nowd^1);
118
119     update(a[now]);
120
121     return now;
122
123   }
124
125   int root;
126   void _build(int L,int R,int nowd) //1-n 的节点 至少为 1
127   {
128     root=build(L,R,nowd,a);
129   }
130
131   int insert(int o,int k,int nowd)
132   {
133     if (!o) return k;
134     int p=a[o].x[nowd];
135     int p2=a[k].x[nowd];
```

```
136       int nx=0;
137       if (p2<=p)
138       {
139         a[o].l=insert(a[o].l,k,nowd^1);
140         nx=a[o].l;
141       }
142       else
143       {
144         a[o].r=insert(a[o].r,k,nowd^1);
145         nx=a[o].r;
146       }
147       update(a[o]);
148
149       if (a[nx].siz>(double)a[o].siz*fac)
150       {
151         pt=0,dfs(o);
152         o=rebuild(1,pt,nowd);
153
154       }
155       return o;
156     }
157     void _insert(int k,int nowd)
158     {
159       int p=root;
160       root = insert(root,k,nowd);
161     }
162
163
164     int _x1,_y1,_x2,_y2;
165     int _ans;
166
167     void ask(int o)
168     {
169       if (o==0) return;
170
171       if (_x1<=a[o].minv[0] && a[o].maxv[0]<=_x2 &&
          ↪  _y1<=a[o].minv[1] && a[o].maxv[1]<=_y2 ) {
172         _ans+=a[o].sumv;return;
173       }
174       if (_x1<=a[o].x[0] && a[o].x[0]<=_x2 && _y1<=a[o].x[1] &&
          ↪  a[o].x[1]<=_y2 ) {
175         _ans+=a[o].w;
176       }
177
178       if (a[o].l) {
179         int p=a[o].l;
```

```
180         if (a[p].minv[0]<=_x2 && _x1<=a[p].maxv[0] &&
            ↪  a[p].minv[1]<=_y2 && _y1<=a[p].maxv[1] )
181           ask(p);
182       }
183       if (a[o].r) {
184         int p=a[o].r;
185         if (a[p].minv[0]<=_x2 && _x1<=a[p].maxv[0] &&
            ↪  a[p].minv[1]<=_y2 && _y1<=a[p].maxv[1] )
186           ask(p);
187       }
188
189
190     }
191
192     int _ask(int x1,int y1,int x2,int y2)
193     {
194       _x1=x1;_y1=y1;_x2=x2;_y2=y2;
195       ;_ans=0;
196       ask(root);
197       return _ans;
198     }
199
200   }S;
201
202   int cmp2(int i,int j){return S.a[i].x[cmp_d]<S.a[j].x[cmp_d];  }
203
204   int main()
205   {
206     int N=read();
207
208     n=0;
209     S.a[++n]=node(N/2,N/2,0);
210     S._build(1,n,0);
211
212     int p;
213     int ans=0;
214     int x,y,A;
215     int x1,y1,x2,y2;
216
217     while (scanf("%d",&p)==1 && p^3)
218     {
219   //    cout<<"t"<<endl;
220       if (p==1) {
221         x=read(),y=read(),A=read();
222         x^=ans;y^=ans;A^=ans;
223         S.a[++n]=node(x,y,A);
```

```
224         S._insert(n,0);
225       } else {
226         x1=read(),y1=read(),x2=read(),y2=read();
227         x1^=ans,y1^=ans,x2^=ans,y2^=ans;
228         ans=S._ask(x1,y1,x2,y2);
229         printf("%d\n",ans);
230       }
231 
232     }
233     return 0;
234   }
```

”

1.27 Kirchhoff

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (998244353)
#define eps (1e-3)
#define MAXN (16+10)
#define MAXM (16*16+10)
typedef __int64 ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}

struct M
{
    int n,m;
    ll a[MAXN][MAXN];
    M(int _n=0){n=m=_n;MEM(a);}
    M(int _n,int _m){n=_n,m=_m;MEM(a);}
    void mem (int _n=0){n=m=_n;MEM(a);}
    void mem (int _n,int _m){n=_n,m=_m;MEM(a);}

  friend M operator*(M a,M b)
    {
```

```
45          M c;
46        For(k,a.m)
47          For(i,a.n)
48                For(j,b.m)
49                    c.a[i][j]=(c.a[i][j]+a.a[i][k]*b.a[k][j])%F;
50      return c;
51      }
52    void make_I(int _n)
53      {
54        n=m=_n; MEM(a)
55          For(i,n) a[i][i]=1;
56      }
57    // 求行列式
58      long double mat[MAXN][MAXN],tmp[MAXN];
59      long double det()
60      {
61        For(i,n) For(j,m) mat[i][j]=a[i][j];
62        For(i,n)
63        {
64          int pos=i;
65          while (fabs(mat[pos][i])<eps&&pos<n) ++pos;
66          if (fabs(mat[pos][i])<eps) continue;
67          if (pos^i)
68          {
69            copy(mat[pos]+1,mat[pos]+1+m+1,tmp+1);
70            copy(mat[i]+1,mat[i]+1+m+1,mat[pos]+1);
71            copy(tmp+1,tmp+1+m+1,mat[i]+1);
72          }
73        For(j,n)
74          if (i^j)
75          {
76            long double p = mat[j][i]/mat[i][i];
77            For(k,m) mat[j][k]-=mat[i][k]*p;
78          }
79        }
80        long double ans=1;
81        For(i,n) ans*=mat[i][i];
82        return ans;
83      }
84  }A,C,D;
85  M pow2(M a,ll b)
86  {
87      M c;c.make_I(a.n);
88      static bool a2[1000000];
89      int n=0;while (b) a2[++n]=b&1,b>>=1;
90      For(i,n)
```

```
91      {
92          if (a2[i]) c=c*a;
93          a=a*a;
94      }
95      return c;
96  }
97
98  const ll
    ↪  p2[]={1,2,4,8,16,32,64,128,256,512,1024,2048,4096,8192,16384,32768,65536};
99
100 M a;
101 int n,m,t[MAXN];
102 void work()
103 {
104   ll ans=0,cnt;
105
106   //t[i] 表示 t 缩点的标号
107   //将 C[G] 的第 a 行, 第 b 列同时去掉后得到的新矩阵 a,b 为任意
    ↪  (1□a,b□n)
108   // 处理 t 最大值为 n-cnt+1
109   a.mem(n-cnt);
110   For(j,n)
111     For(l,n)
112       if (t[j]!=t[l]&&A.a[j][l])
113       {
114         a.a[t[j]][t[j]]++;
115         a.a[t[j]][t[l]]--;
116       }
117   ll t2=(ll)(fabs(a.det())+eps)%F;
118
119   cout<<ans<<endl;
120 }
121
122 int u[MAXN],v[MAXN];
123 void Kirchhoff()
124 {
125   while (cin>>n>>m) {
126     A.mem(n),D.mem(n),C.mem(n);
127     For(i,m)
128     {
129       scanf("%d%d",&u[i],&v[i]);
130       D.a[u[i]][u[i]]++;
131       D.a[v[i]][v[i]]++;
132       A.a[u[i]][v[i]]++;
133       A.a[v[i]][u[i]]++;
```

```
134       }
135       work();
136     }
137
138   }
139
140
141   int main()
142   {
143   //  freopen(".in","r",stdin);
144
145
146
147
148
149     return 0;
150   }
```

”

1.28 KMP

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (1000000+10)
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
// kmp
class KMP
{
public:
  int f2[MAXN]; //字符串从 0 开始，但是 f[i] 表示匹配第 i 个字符，
  ↪ 前面留一个 f[0]--a-->f[1]--... 这样的
  char T2[MAXN],P2[MAXN]; //T is long,P is model str
  void mem(){MEM(f2) MEM(T2) MEM(P2)  }
  int getFail(char *P=0,int* f=0)
  {
    if (P==0) P=P2;if (f==0) f=f2;
    int m=strlen(P);
    f[0]=f[1]=0;
    For(i,m-1)
```

```
44      {
45        int j=f[i];
46        while(j&&P[i]!=P[j]) j=f[j];
47        f[i+1]= P[i] == P[j] ? j+1 : 0;
48      }
49    }
50    int find(char* T=0,char* P=0,int* f=0)
51    {
52      if (T==0) T=T2;if (P==0) P=P2;if (f==0) f=f2;
53      int n=strlen(T),m=strlen(P);
54      getFail(P,f);
55      int j=0;
56      Rep(i,n)
57      {
58        while(j&&T[i]!=P[j]) j=f[j];
59        if (T[i]==P[j]) j++;
60        if (j==m) return i-m+1;
61      }
62    }
63  }S;
64  int main()
65  {
66    freopen(".in","r",stdin);
67  //  freopen(".out","w",stdout);
68
69
70    return 0;
71  }
```

”

1.29 kosaraju

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,0x3f,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEMx(a,b) memset(a,b,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (100000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
```

```
  return x*f;
}
#define MAXN (20000+10)
vi G[MAXN],G2[MAXN];
vi S;
int vis[MAXN],sccno[MAXN],scc_cnt;
void dfs1(int u) {
  if (vis[u]) return ;
  vis[u] = 1;
  int sz=SI(G[u]);
  Rep(i,sz) dfs1(G[u][i]);
  S.pb(u);
}
void dfs2(int u) {
  if (sccno[u]) return;
  sccno[u] = scc_cnt;
  int sz=SI(G2[u]);
  Rep(i,sz) dfs2(G2[u][i]);
}
void find_scc(int n) {
  scc_cnt=0;
  S.clear();
  MEM(sccno) MEM(vis)
  Rep(i,n) dfs1(i);
  RepD(i,n-1)
    if (!sccno[S[i]]) {
      scc_cnt++; dfs2(S[i]);
    }
}
int main()
{
//  freopen(".in","r",stdin);
//  freopen(".out","w",stdout);

  int T=read();
  while(T--) {
    int n=read(),m=read();
    Rep(i,m) {
      int u=read()-1,v=read()-1;
      G[u].pb(v);G2[v].pb(u);
    }
    find_scc(n);
  }

  return 0;
}
```

”

1.30 LCT

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (200000+10)
#define MAXM (100000+10)
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}

int n,m;

class Splay
{
public:
  int father[MAXN],siz[MAXN];
  int ch[MAXN][2];
  bool root[MAXN];
  void mem(int n)
  {
    MEM(father) MEM(siz) MEM(root)
    For(i,n+1) siz[i]=1,root[i]=1;root[0]=1;
```

```
45      MEM(ch)
46
47    }
48    void maintain(int x)
49    {
50      siz[x]=siz[ch[x][0]]+siz[ch[x][1]]+1;
51    }
52    void rotate(int x)
53    {
54      int y=father[x],kind=ch[y][1]==x;
55      ch[y][kind]=ch[x][!kind];
56      if (ch[y][kind]) {
57        father[ch[y][kind]]=y;
58      }
59      father[x]=father[y];
60      father[y]=x;
61      ch[x][!kind]=y;
62      if (root[y])
63      {
64        root[x]=1;root[y]=0;
65      }
66      else
67      {
68        ch[father[x]][ ch[father[x]][1]==y ] = x;
69      }
70      maintain(y);maintain(x);
71    }
72    void splay(int x)
73    {
74      while(!root[x])
75      {
76        int y=father[x];
77        int z=father[y];
78        if (root[y]) rotate(x);
79        else if ( (ch[y][1]==x)^(ch[z][1]==y) )
80        {
81          rotate(x); rotate(x);
82        }
83        else
84        {
85          rotate(y); rotate(x);
86        }
87      }
88    }
89
90    int access(int x)
```

```
91    {
92      int y=0;
93      do
94      {
95        splay(x);
96        if (ch[x][1]) root[ch[x][1]]=1;
97        ch[x][1]=y;
98        if (y) root[y]=0;
99        maintain(x);
100       y = x;
101       x=father [x];
102     } while (x) ;
103     return y;
104   }
105   void cut(int x)
106   {
107     access(x);
108     splay(x);
109
110     father[ch[x][0]]=0;
111     root[ch[x][0]]=1;
112     ch[x][0]=0;
113     maintain(x);
114   }
115
116   void join(int x,int w)
117   {
118     father[x]=w;
119   }
120   int find_root(int x) {
121     access(x);
122     splay(x);
123     int t=x;
124     while(ch[t][0]) t=ch[t][0];
125     return t;
126   }
127
128 }S;
129
130
131 int main()
132 {
133 //  freopen(".in","r",stdin);
134 //  freopen(".out","w",stdout);
135
136
```

```
137
138     return 0;
139 }
```

”

1.31 LCT2

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (10000+10)
#define MAXM (200000+10)
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}

int n,m;

class Splay
{
public:
  int father[MAXN],siz[MAXN];
  int ch[MAXN][2];
  bool root[MAXN];
  bool rev[MAXN];
  void mem(int n)
  {
    MEM(father) MEM(siz) MEM(root)
```

```
45      For(i,n+1) siz[i]=1,root[i]=1;root[0]=1;
46      MEM(ch)  MEM(rev)
47    }
48    void pushdown(int x)
49    {
50      if (!x) return ;
51      if (rev[x]) {
52        if (ch[x][0]) rev[ch[x][0]]^=1;
53        if (ch[x][1]) rev[ch[x][1]]^=1;
54        swap(ch[x][0],ch[x][1]);
55        rev[x]^=1;
56      }
57    }
58
59    void maintain(int x)
60    {
61      siz[x]=siz[ch[x][0]]+siz[ch[x][1]]+1;
62    }
63    void rotate(int x)
64    {
65      int y=father[x],kind=ch[y][1]==x;
66
67      ch[y][kind]=ch[x][!kind];
68      if (ch[y][kind]) {
69        father[ch[y][kind]]=y;
70      }
71      father[x]=father[y];
72      father[y]=x;
73      ch[x][!kind]=y;
74      if (root[y])
75      {
76        root[x]=1;root[y]=0;
77      }
78      else
79      {
80        ch[father[x]][ ch[father[x]][1]==y ] = x;
81      }
82      maintain(y);maintain(x);
83    }
84
85    void P(int x)
86    {
87      if (!root[x]) P(father[x]);
88      pushdown(x);
89    }
90
```

```
91  void splay(int x)
92  {
93    P(x);
94    while(!root[x])
95    {
96      int y=father[x];
97      int z=father[y];
98      if (root[y]) rotate(x);
99      else if ( (ch[y][1]==x)^(ch[z][1]==y) )
100     {
101       rotate(x); rotate(x);
102     }
103     else
104     {
105       rotate(y); rotate(x);
106     }
107   }
108 }
109
110
111
112 int access(int x)
113 {
114   int y=0;
115   do
116   {
117     splay(x);
118     if (ch[x][1]) root[ch[x][1]]=1;
119     ch[x][1]=y;
120     if (y) root[y]=0;
121     maintain(x);
122     y = x;
123     x=father [x];
124   } while (x) ;
125   return y;
126 }
127
128 void cut(int x)
129 {
130   access(x);
131   splay(x);
132
133   father[ch[x][0]]=0;
134   root[ch[x][0]]=1;
135   ch[x][0]=0;
136   maintain(x);
```

```
137   }
138   int find_root(int x) {
139     access(x);
140     splay(x);
141     int t=x;
142     while(ch[t][0]) t=ch[t][0];
143     return t;
144   }
145
146   void join(int x,int y)
147   {
148     make_root(x);
149     access(y);
150     splay(y);
151     ch[y][1]=x;
152     father[x]=y;
153     root[x]=0;
154   }
155   void reverse(int x){
156     rev[x]^=1;   // 标记记完后迅速处理
157   }
158   void make_root(int x){
159     access(x);splay(x);
160     reverse(x);pushdown(x);
161   }
162   int get_root(int x){
163     access(x);
164     splay(x);
165     do {
166       pushdown(x);
167       if (ch[x][0]) x=ch[x][0];
168       else break;
169     }while(1);
170     return x;
171   }
172
173   bool check(int x,int y) {
174     while (father[x]) x=father[x];
175     while (father[y]) y=father[y];
176     return x==y;
177   }
178
179 }S;
180
181 int main()
182 {
```

```
183  //  freopen(".in","r",stdin);
184  //  freopen(".out","w",stdout);
185
186
187
188    return 0;
189  }
```

”

1.32 LCT_ 链上修改

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (51061)
#define MAXN (100000+10)
#define MAXQ (100000+10)
#define MAXC (10000)
typedef unsigned int ll;
void mul(ll &a,ll b){a=(a*b)%F;}
void add(ll &a,ll b){a=(a%F+b%F)%F;}


class LCT
{
public:
  int father[MAXN],siz[MAXN];
  int ch[MAXN][2];
  bool root[MAXN];
  bool rev[MAXN];
  ll addv[MAXN],mulv[MAXN],sumv[MAXN],val[MAXN];
  void mem(int n)
  {
    MEM(father) MEM(siz) MEM(root)
```

```
44      For(i,n+1) siz[i]=root[i]=mulv[i]=val[i]=sumv[i]=1;
        ↪  root[0]=1;
45      MEM(ch)  MEM(rev)
46      MEM(addv)
47    }
48    void pushdown(int x)
49    {
50      if (!x) return ;
51      if (rev[x]) {
52        if (ch[x][0]) rev[ch[x][0]]^=1;
53        if (ch[x][1]) rev[ch[x][1]]^=1;
54        swap(ch[x][0],ch[x][1]);
55        rev[x]^=1;
56      }
57      if (mulv[x]!=1) {
58        if (ch[x][0])
          ↪  mul(mulv[ch[x][0]],mulv[x]),mul(addv[ch[x][0]],mulv[x]),mul(val[ch[x][0]],mulv[x]),
59        if (ch[x][1])
          ↪  mul(mulv[ch[x][1]],mulv[x]),mul(addv[ch[x][1]],mulv[x]),mul(val[ch[x][1]],mulv[x]),
60        mulv[x]=1;
61      }
62      if (addv[x]) {
63        if (ch[x][0])
          ↪  add(addv[ch[x][0]],addv[x]),add(val[ch[x][0]],addv[x]),add(sumv[ch[x][0]],addv[x]*s
64        if (ch[x][1])
          ↪  add(addv[ch[x][1]],addv[x]),add(val[ch[x][1]],addv[x]),add(sumv[ch[x][1]],addv[x]*s
65        addv[x]=0;
66      }
67    }
68    void maintain(int x)
69    {
70      siz[x]=siz[ch[x][0]]+siz[ch[x][1]]+1;
71      sumv[x]=(sumv[ch[x][0]]+sumv[ch[x][1]]+val[x])%F;
72    }
73    void rotate(int x)
74    {
75      int y=father[x],kind=ch[y][1]==x;
76
77      ch[y][kind]=ch[x][!kind];
78      if (ch[y][kind]) {
79        father[ch[y][kind]]=y;
80      }
81      father[x]=father[y];
82      father[y]=x;
83      ch[x][!kind]=y;
84      if (root[y])
```

```
85      {
86        root[x]=1;root[y]=0;
87      }
88      else
89      {
90        ch[father[x]][ ch[father[x]][1]==y ] = x;
91      }
92      maintain(y);maintain(x);
93    }
94
95    void P(int x)
96    {
97      if (!root[x]) P(father[x]);
98      pushdown(x);
99    }
100
101   void splay(int x)
102   {
103     P(x);
104     while(!root[x])
105     {
106       int y=father[x];
107       int z=father[y];
108       if (root[y]) rotate(x);
109       else if ( (ch[y][1]==x)^(ch[z][1]==y) )
110       {
111         rotate(x); rotate(x);
112       }
113       else
114       {
115         rotate(y); rotate(x);
116       }
117     }
118   }
119
120
121
122   int access(int x)
123   {
124     int y=0;
125     do
126     {
127       splay(x);
128       if (ch[x][1]) root[ch[x][1]]=1;
129       ch[x][1]=y;
130       if (y) root[y]=0;
```

```
131         maintain(x);
132         y = x;
133         x=father [x];
134       } while (x) ;
135       return y;
136     }
137
138     void cut(int x)
139     {
140       access(x);
141       splay(x);
142
143       father[ch[x][0]]=0;
144       root[ch[x][0]]=1;
145       ch[x][0]=0;
146       maintain(x);
147     }
148
149     void join(int x,int y)
150     {
151       make_root(x);
152       access(y);
153       splay(y);
154       ch[y][1]=x;
155       father[x]=y;
156       maintain(y);
157       root[x]=0;
158     }
159     void reverse(int x){
160       rev[x]^=1;   // 标记记完后迅速处理
161     }
162     void make_root(int x){
163       access(x);splay(x);
164       reverse(x);pushdown(x);
165     }
166     int get_root(int x){
167       access(x);
168       splay(x);
169       do {
170         pushdown(x);
171         if (ch[x][0]) x=ch[x][0];
172         else break;
173       }while(1);
174       return x;
175     }
176
```

```
177   void Mul(int x,ll cost){
178     pushdown(x);mulv[x]=cost;
179     mulv[x]=cost;mul(val[x],cost);mul(addv[x],cost);//mul(sumv[x],cost);
        ↪
180   }
181   void Add(int x,ll cost){
182     pushdown(x);
183     addv[x]=cost;add(val[x],cost);//add(sumv[x],cost*siz[x]);
184   }
185
186 }S;
187
188 int n,q;
189
190 int main()
191 {
192 //  freopen(".in","r",stdin);
193 //  freopen(".out","w",stdout);
194
195   scanf("%d%d",&n,&q);
196   S.mem(n);
197   For(i,n-1) {
198     int u,v;
199     scanf("%d%d",&u,&v);
200     S.join(u,v);
201   }
202
203   For(i,q) {
204     char c[2];
205     int u,v;
206     scanf("%s%d%d",c,&u,&v);
207     if (c[0]=='+'||c[0]=='*') {
208       int cost;
209       scanf("%d",&cost);
210       S.make_root(u);
211       S.access(v);
212       S.splay(v);
213       if ('+'==c[0]) S.Add(v,cost);
214       else S.Mul(v,cost);
215
216     } else if (c[0]=='-') {
217       int u2,v2;
218       scanf("%d%d",&u2,&v2);
219       S.make_root(u);
220       S.cut(v);
221       S.join(u2,v2);
```

```
222
223       } else if (c[0]=='/') {
224         S.make_root(u);
225         S.access(v);
226         S.splay(v);
227         printf("%u\n",S.sumv[v]%F);
228       }
229     }
230
231
232
233     return 0;
234   }
```

”

1.33 Linked_List

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
typedef long long ll;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
#define MAXN (500000+10)
struct link
{
    int pre,next;
}l[MAXN];
void del(int x)
{
```

```
    l[l[x].pre].next=l[x].next;
    l[l[x].next].pre=l[x].pre;
}
void del(int x,int y)  //要求 x,y 同时在链表中
{
  if (x>y) swap(x,y);
    l[l[x].pre].next=l[y].next;
    l[l[y].next].pre=l[x].pre;
}
int n=100;
int main()
{
//  freopen(".in","r",stdin);
//  freopen(".out","w",stdout);

  for (int i=1;i<=n;i++) {l[i-1].next=i;l[i].pre=i-1;}
   ↪  l[n].next=0;

  for(int i=l[0].next;i;i=l[i].next) cout<<i<<' ';


  return 0;
}
```

”

1.34 link_table

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (100000+10)
#define MAXM (60000*2+10)
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}

class SPFA
{
public:
  void mem()
  {
    MEM(pre) MEM(edge) MEM(pre) MEM(weight) size=1;
  }
  int q[MAXN*100];
  int edge[MAXM],next[MAXM],pre[MAXN],weight[MAXM],size;
  void addedge(int u,int v,int w)
  {
    edge[++size]=v;
```

```
45      weight[size]=w;
46      next[size]=pre[u];
47      pre[u]=size;
48    }
49    void addedge2(int u,int v,int w){addedge(u,v,w);addedge(v,u,w);}
50    int d[MAXN];
51    bool b[MAXN];
52    int spfa(int s,int t)
53    {
54      MEM(b) MEMI(d)
55      b[s]=1; d[s]=0;
56
57      int head=1,tail=1;q[1]=1;
58      while(head<=tail)
59      {
60        int now=q[head++];
61        b[now]=0;
62        Forp(now)
63        {
64          int v=edge[p];
65          if (d[now]+weight[p]<d[v]) {
66            d[v]=d[now]+weight[p];
67            if (!b[v]) { b[v]=1,q[++tail]=v;
68            }
69          }
70        }
71      }
72      return d[t];
73    }
74  }S1,S2;
75  class link_table
76  {
77  public:
78    void mem()
79    {
80      MEM(pre) MEM(edge) MEM(pre) MEM(weight) size=1;
81    }
82    int q[MAXN*100];
83    int edge[MAXM],next[MAXM],pre[MAXN],weight[MAXM],size;
84
85    void addedge(int u,int v,int w)
86    {
87      edge[++size]=v;
88      weight[size]=w;
89      next[size]=pre[u];
90      pre[u]=size;
```

```
91    }
92    void addedge2(int u,int v,int w){addedge(u,v,w);addedge(v,u,w);}
93  }St;
94  int n,m;
95  int main()
96  {
97  //  freopen(".in","r",stdin);
98  //  freopen(".out","w",stdout);
99
100   return 0;
101 }
```

”

1.35 make_prime

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define MAXN (100000+10)
typedef long long ll;
int p[MAXN],tot;
bool b[MAXN]={0};
void make_prime(int n)
{
  tot=0;
  Fork(i,2,n)
  {
    if (!b[i]) p[++tot]=i;
    For(j,tot)
    {
      if (i*p[j]>n) break;
      b[i*p[j]]=1;
      if (i%p[j]==0) break;
    }
  }
}
int main()
{

```

```
45
46      return 0;
47  }
```

”

1.36 make_prime_mul

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define MAXN (100000+10)
typedef long long ll;
int p[MAXN],tot;
bool b[MAXN]={0};
int mul[MAXN];
ll i2muli[MAXN],s[MAXN];
void make_prime(int n)
{
    tot=0; mul[1]=1;
    Fork(i,2,n)
    {
        if (!b[i]) p[++tot]=i,mul[i]=-1;
        For(j,tot)
        {
            if (i*p[j]>n) break;
            b[i*p[j]]=1;
            mul[i*p[j]]=-mul[i];
            if (i%p[j]==0) {
                mul[i*p[j]]=0;
                break;
            }
```

```
45          }
46      }
47      For(i,n) i2muli[i]=(ll)i*i%modp*mul[i];
48      s[0]=0;
49      For(i,n) s[i]=(s[i-1]+i2muli[i]+modp)%modp;
50  }
51  int main()
52  {
53
54
55    return 0;
56  }
```

”

1.37 make_prime_phi

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define MAXN (4000000+10)
typedef long long ll;
typedef unsigned long long ull;
int p[MAXN],tot,phi[MAXN];
bool b[MAXN]={0};
void make_prime(int n)
{
  tot=0; phi[1]=1;
  Fork(i,2,n)
  {
    if (!b[i]) p[++tot]=i,phi[i]=i-1;
    For(j,tot)
    {
      if (i*p[j]>n) break;
      b[i*p[j]]=1;
      phi[i*p[j]]=phi[i]*phi[p[j]];
      if (i%p[j]==0) {
        phi[i*p[j]]= phi[i]*p[j];
        break;
      }
    }
```

```
45    }
46  }
47  int main()
48  {
49    return 0;
50  }
```

”

1.38 manacher

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (10000+10)
#define Sp_char1 ('*')
#define Sp_char2 ('$')
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
class manacher
{
public:
  int n;
  char s[MAXN];
  int p[2*MAXN+2];
  manacher(){n=0; MEM(s) MEM(p)}
  manacher(char *_s){n=0; MEM(s) if (_s)
  ↪  memcpy(s,_s,sizeof(char)*(strlen(_s)+1)),n=strlen(s);
  ↪  MEM(p)}
```

```
41    void mem(char *_s){n=0; MEM(s) if (_s)
      ↪  memcpy(s,_s,sizeof(char)*(strlen(_s)+1)),n=strlen(s);
      ↪  MEM(p)}
42    char str[MAXN*2+2];
43    void work()
44    {
45      str[0]=Sp_char1;
46      Rep(i,n) str[2*i+1]=Sp_char2,str[2*i+2]=s[i];
47      str[2*n+1]=Sp_char2; str[2*n+2]='\0';
48
49      n=2*n+2; MEM(p)
50      int mx=0,id=0;
51      For(i,n-1)
52      {
53        if (i<mx) p[i]=min(p[2*id-i],mx-i);
54
55        while(str[i-p[i]]==str[i+p[i]]) ++p[i];
56        if (mx<i+p[i]) //mx 为已查明的最右端
57        {
58          mx=i+p[i];
59          id=i;
60        }
61      }
62    }
63  }S;
64  int main()
65  {
66  //  freopen(".in","r",stdin);
67  //  freopen(".out","w",stdout);
68
69    return 0;
70  }
```

”

1.39 Math

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<vector>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %I64d\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
#define MAXN (1000000)
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+(a-b)/F*F+F)%F;}
```

```
45  void upd(ll &a,ll b){a=(a%F+b%F)%F;}
46  char s[]="no solution\n";
47  class Math
48  {
49  public:
50    ll gcd(ll a,ll b){if (!b) return a;return gcd(b,a%b);}
51    void gcd(ll a,ll b,ll &d,ll &x,ll &y) {
52      if (!b) {d=a,x=1,y=0;  }
53      else {gcd(b,a%b,d,y,x); y-=x*(a/b);  }
54    }
55    // x=a[i](mod m[i]) (0<=i<n)
56     ll china(int n,int *a,const int *m) {
57          ll M=F-1,d,y,x=0,ans=0;
58          Rep(i,n) {
59              ll w=M/m[i];
60              gcd(w,m[i],d,x,y);
61              ans=(ans+(ll)x*w*a[i])%M;
62          }
63          return (ans+M)%M;
64      }
65    ll abs(ll x){if (x>=0) return x;return -x;}
66    ll exgcd(ll a,ll b,ll &x, ll &y)
67    {
68        if (!b) {x=1,y=0;return a;}
69        ll g=exgcd(b,a%b,x,y);
70        ll t=x;x=y;y=t-a/b*y;
71        return g;
72    }
73    ll pow2(ll a,int b,ll p)  //a^b mod p
74    {
75        if (b==0) return 1%p;
76        if (b==1) return a%p;
77        ll c=pow2(a,b/2,p)%p;
78        c=c*c%p;
79        if (b&1) c=c*a%p;
80        return c%p;
81    }
82    ll inv(ll a,ll p) { //gcd(a,p)=1
83      return pow2(a,p-2,p);
84    }
85    ll factnmodp(ll n,ll p) {
86      ll t=n;
87      while(n) {
88        t=(t-n%p+F)%F;
89        n/=p;
90      }
```

```
  return t*inv(p-1,p);
}
ll get_factor(vector<ll> &v,ll p) {
  for(ll i=2;i*i<=p;i++) if (p%i==0) {
    v.pb(i);
    if (i*i<p) v.pb(p/i);
  }
  sort(v.begin(),v.end());
}
template <class T>
ll find(vector<T> v,T x) {
  return lower_bound(v.begin(),v.end(),x)-v.begin();
}
// p is prime
// certainly their are phi(p) root
ll get_primitiveRoot(ll p) {
  p;
  vector<ll> v;
  get_factor(v,p-1);
  for(ll i=2;;i++) {
    bool fl=0;
    Rep(j,SI(v)) {
      if (pow2(i,v[j],p)==1) {
        fl=1; break;
      }
    }
    if (!fl) return i;
  }
}
ll Modp(ll a,ll b,ll p)  //a*x=b (mod p)
{
    ll x,y;
    ll g=exgcd(a,p,x,y),d;
    if (b%g) {return -1;}
    d=b/g;x*=d,y*=d;
    x=(x+abs(x)/p*p+p)%p;
    return x;
}
int h[MAXN];
ll hnum[MAXN];
int hash(ll x)
{
    int i=x%MAXN;
    while (h[i]&&hnum[i]!=x) i=(i+1)%MAXN;
    hnum[i]=x;
    return i;
```

```
  }
  ll babystep(ll a,ll b,int p)  // a^x = b (mod p)
  {
    MEM(h) MEM(hnum)
    int m=sqrt(p);while (m*m<p) m++;
      ll res=b,ans=-1;
      ll uni=pow2(a,m,p);
      if (!uni) if (!b) ans=1;else ans=-1;
      else
      {
          Rep(i,m+1)
          {
              int t=hash(res);
              h[t]=i+1;
              res=(res*a)%p;
          }
          res=uni;
      For(i,m+1)
      {
              int t=hash(res);
              if (h[t]) {ans=i*m-(h[t]-1);break;}else hnum[t]=0;
              res=res*uni%p;
          }
    }
    return ans;
  }


}S;

int main()
{
//  freopen(".in","r",stdin);
//  freopen(".out","w",stdout);



  return 0;
}
```

”

1.40 Matrix

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (998244353)
#define eps (1e-3)
#define MAXN (16+10)
#define MAXM (16*16+10)
typedef __int64 ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}

typedef double Matrix[MAXN][MAXN];

void gauss_elimination(Matrix A, int n) { //假设系数矩阵 A 可逆
↪  A[0..n-1,0..n]
//运行结束后 A[i][i] 极为第 i 个变量的值
  Rep(i,n) {
    int r=i;
    Fork(j,i+1,n-1) {
      if (fabs(A[j][i])>fabs(A[r][i])) r=j;
    }
```

```
    if (r>i) {
      Rep(j,n+1) swap(A[r][j],A[i][j]);
    }
    /* 不精确
    Fork(k,i+1,n-1) {
      double f = A[k][i] / A[i][i];
      Fork(j,i,n) A[k][j] -= f * A[i][j];
    }*/

    Fork(k,i+1,n-1) {
      ForkD(j,i,n) A[k][j] -= A[k][i] / A[i][i] * A[i][j];
    }

  }
  RepD(i,n-1) {
    Fork(j,i+1,n-1) A[i][n] -= A[j][n] * A[i][j];
    A[i][n] /= A[i][i];
  }
}

void gauss_jordan(Matrix A, int n) { //矛盾方程和多余方程都可以
↪  A[0..n-1,0..n]
//运行结束后 A[i][i] 极为第 i 个变量的值
  Rep(i,n) {
    int r=i;
    Fork(j,i+1,n-1) {
      if (fabs(A[j][i])>fabs(A[r][i])) r=j;
    }
    if (fabs(A[r][i]) < eps ) continue;
    if (r>i) {
      Rep(j,n+1) swap(A[r][j],A[i][j]);
    }

    Rep(k,n) if (k^i) {
      double f = A[k][i] / A[i][i];
      ForkD(j,i,n) A[k][j] -= f * A[i][j];
    }
  }
}

struct M
{
    int n,m;
    ll a[MAXN][MAXN];
    M(int _n=0){n=m=_n;MEM(a);}
    M(int _n,int _m){n=_n,m=_m;MEM(a);}
```

```
89      void mem (int _n=0){n=m=_n;MEM(a);}
90      void mem (int _n,int _m){n=_n,m=_m;MEM(a);}
91
92    friend M operator*(M a,M b)
93      {
94          M c(a.n,b.m);
95        For(k,a.m)
96          For(i,a.n)
97                For(j,b.m)
98                    c.a[i][j]=(c.a[i][j]+a.a[i][k]*b.a[k][j])%F;
99      return c;
100     }
101     friend M operator+(M a,M b)
102     {
103       For(i,a.n)
104             For(j,a.m)
105                 a.a[i][j]=(a.a[i][j]+b.a[i][j])%F;
106     return a;
107     }
108   void make_I(int _n)
109     {
110       n=m=_n; MEM(a)
111         For(i,n) a[i][i]=1;
112     }
113   // 求行列式
114     long double mat[MAXN][MAXN],tmp[MAXN];
115     long double det()
116     {
117       For(i,n) For(j,m) mat[i][j]=a[i][j];
118       For(i,n)
119       {
120         int pos=i;
121         while (fabs(mat[pos][i])<eps&&pos<n) ++pos;
122         if (fabs(mat[pos][i])<eps) continue;
123         if (pos^i)
124         {
125           copy(mat[pos]+1,mat[pos]+1+m+1,tmp+1);
126           copy(mat[i]+1,mat[i]+1+m+1,mat[pos]+1);
127           copy(tmp+1,tmp+1+m+1,mat[i]+1);
128         }
129       For(j,n)
130         if (i^j)
131         {
132           long double p = mat[j][i]/mat[i][i];
133           For(k,m) mat[j][k]-=mat[i][k]*p;
134         }
```

```
135         }
136         long double ans=1;
137         For(i,n) ans*=mat[i][i];
138         return ans;
139     }
140 }A,C,D;
141 M pow2(M a,ll b)
142 {
143     M c;c.make_I(a.n);
144     static bool a2[1000000];
145     int n=0;while (b) a2[++n]=b&1,b>>=1;
146     For(i,n)
147     {
148         if (a2[i]) c=c*a;
149         a=a*a;
150     }
151     return c;
152 }
153 bool a3[1000000];
154 M pow222(M a,ll b)
155 {
156     M c;c.make_I(a.n);
157     int n=0;while (b) a3[++n]=b&1,b>>=1;
158     c=a; b=1;
159     M d=c;
160   ForD(i,n-1)
161     {
162     b=b*2+a3[i];
163       c=c*d+c;
164       d=d*d;
165       if (a3[i]) c=c*a+a,d=d*a;
166     }
167     return c;
168 }
169
170
171
172
173 const ll
    ↪  p2[]={1,2,4,8,16,32,64,128,256,512,1024,2048,4096,8192,16384,32768,65536};
174 M a;
175 int n,m,t[MAXN];
176 void work()
177 {
178   ll ans=0,cnt;
179
```

```
180   //t[i] 表示 t 缩点的标号
181   //将 C[G] 的第 a 行，第 b 列同时去掉后得到的新矩阵 a,b 为任意
      ↪ (1≤a,b≤n)
182   // 处理 t 最大值为 n-cnt+1
183   a.mem(n-cnt);
184   For(j,n)
185     For(l,n)
186       if (t[j]!=t[l]&&A.a[j][l])
187       {
188         a.a[t[j]][t[j]]++;
189         a.a[t[j]][t[l]]--;
190       }
191   ll t2=(ll)(fabs(a.det())+eps)%F;
192
193   cout<<ans<<endl;
194 }
195
196 int u[MAXN],v[MAXN];
197 void Kirchhoff()
198 {
199   while (cin>>n>>m) {
200     A.mem(n),D.mem(n),C.mem(n);
201     For(i,m)
202     {
203       scanf("%d%d",&u[i],&v[i]);
204       D.a[u[i]][u[i]]++;
205       D.a[v[i]][v[i]]++;
206       A.a[u[i]][v[i]]++;
207       A.a[v[i]][u[i]]++;
208     }
209     work();
210   }
211
212 }
213
214
215 int main()
216 {
217 //  freopen(".in","r",stdin);
218
219
220
221
222
223   return 0;
```

```
224     }
```

”

1.41 Max_flow

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (1000000+100)
#define MAXM (6000000+100)
long long mul(long long a,long long b){return (a*b)%F;}
long long add(long long a,long long b){return (a+b)%F;}
long long sub(long long a,long long b){return
 ↪  (a-b+(a-b)/F*F+F)%F;}
typedef long long ll;
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
class Max_flow  //dinic+ 当前弧优化
{
public:
    int n,t;
    int q[MAXN];
    int edge[MAXM],Next[MAXM],Pre[MAXN],weight[MAXM],size;
```

```
44      void addedge(int u,int v,int w)
45      {
46          edge[++size]=v;
47          weight[size]=w;
48          Next[size]=Pre[u];
49          Pre[u]=size;
50      }
51      void addedge2(int u,int v,int
        ↪  w){addedge(u,v,w),addedge(v,u,0);}
52      bool b[MAXN];
53      int d[MAXN];
54      bool SPFA(int s,int t)
55      {
56          For(i,n) d[i]=INF;
57          MEM(b)
58          d[q[1]=s]=0;b[s]=1;
59          int head=1,tail=1;
60          while (head<=tail)
61          {
62              int now=q[head++];
63              Forp(now)
64              {
65                  int &v=edge[p];
66                  if (weight[p]&&!b[v])
67                  {
68                      d[v]=d[now]+1;
69                      b[v]=1,q[++tail]=v;
70                  }
71              }
72          }
73          return b[t];
74      }
75      int iter[MAXN];
76      int dfs(int x,int f)
77      {
78          if (x==t) return f;
79          Forpiter(x)
80          {
81              int v=edge[p];
82              if (weight[p]&&d[x]<d[v])
83              {
84                    int nowflow=dfs(v,min(weight[p],f));
85                    if (nowflow)
86                    {
87                      weight[p]-=nowflow;
88                      weight[p^1]+=nowflow;
```

```
                    return nowflow;
                }
            }
        }
        return 0;
    }
    int max_flow(int s,int t)
    {
        (*this).t=t;
        int flow=0;
        while(SPFA(s,t))
        {
            For(i,n) iter[i]=Pre[i];
            int f;
            while (f=dfs(s,INF))
                flow+=f;
        }
        return flow;
    }
    void mem(int n)
    {
        (*this).n=n;
        size=1;
        MEM(Pre)
    }
}S;
int main()
{
//  freopen(".in","r",stdin);
//  freopen(".out","w",stdout);


  return 0;
}
```

”

1.42 merge_count

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
#include<vector>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (10000001)
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}

int A[MAXN],t[MAXN];
int fl=0;
ll merge_count(int l,int r)
{
  int n=r-l+1,m=(l+r)>>1;
  if (n<=1) return 0;

  ll cnt=0;
  cnt+=merge_count(l,m);
  cnt+=merge_count(m+1,r);

  int p=l,q=m+1,ai=l;
```

```
  while (ai<=r)
  {
    if ( q>r || (p<=m && A[p]<=A[q] ) )
    {
      t[ai++]=A[p++];
    }
    else {
      cnt+=m-p+1;
      t[ai++]=A[q++];
    }
  }
  Fork(i,l,r) A[i]=t[i];
  return cnt;
}
int n=10000000;
int main()
{
//  freopen(".in","r",stdin);
//  freopen(".out","w",stdout);

  For(i,n) A[i]=n-i;
  cout<<merge_count(1,n);

  return 0;
}
```

”

1.43 model

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int main()
{
  freopen(".in","r",stdin);
  freopen(".out","w",stdout);



  return 0;
}
```

”

1.44 Palindromic_Tree

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define pb push_back
#define mp make_pair
#define MAXN (600000+10)
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}

namespace Palindromic_Tree {
  int s[MAXN],n;
  int tot,next[MAXN][26],link[MAXN],len[MAXN],last;
  int cnt[MAXN];

  int newnode(int l)
  {
    len[tot]=l;
    return tot++;
  }
  void mem() {
    MEM(s) MEM(next) MEM(link) MEM(len) MEM(cnt)
    n=tot=0;
    newnode(0); newnode(-1);
    link[0]=link[1]=1; s[0]=27;
    last=0;
  }

```

```
45    int getnode(int x)
46    {
47      while (s[ n - len[x]-1 ]  != s[n] ) x=link[x];
48      return x;
49    }
50
51    void add(int c) {
52      s[++n]=c;
53  //    cout<<"1";
54      int cur=getnode(last);
55      if (!next[cur][c])
56      {
57        int now=newnode(len[cur]+2);
58        int tmp=getnode(link[cur]);
59        link[now]=next[tmp][c];
60        next[cur][c] = now;
61
62      }
63      last=next[cur][c];
64      cnt[last]++;
65    }
66
67    void work()
68    {
69      RepD(i,tot) cnt[link[i]]+=cnt[i];
70
71      ll ans=0;
72      Fork(i,2,tot) {
73        ans=max(ans,1LL*cnt[i]*len[i]);
74
75      }
76      cout<<ans<<endl;
77
78    }
79
80  }
81  using namespace Palindromic_Tree;
82
83  char S[MAXN];
84  int N;
85  int main()
86  {
87  //  freopen(".in","r",stdin);
88  //  freopen(".out","w",stdout);
89
90    Palindromic_Tree::mem();
```

```
91
92    scanf("%s",S);
93    int N=strlen(S);
94    Rep(i,N) Palindromic_Tree::add(S[i]-'a');
95
96    Palindromic_Tree::work();
97
98    return 0;
99  }
```

”

1.45 Partition_of_k

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
#include<iomanip>
#include<vector>
#include<string>
#include<queue>
#include<stack>
#include<map>
#include<sstream>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (1000000007 )
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %I64d\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
```

```
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F+2*F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
#define MAXN (100000+10)
ll tar[MAXN];
void prework() {
  MEM(tar)
  tar[0] = 1;
  For(i,1e5) {
    tar[i]=0;
    for(int k=1;;k++) {
      int b=(k&1)?1:-1 ;
      int j1=i - k*(3*k-1)/2;
      int j2=i - k*(3*k+1)/2;
      if (j1<0) break;
      if (j1>=0) upd(tar[i],b*tar[j1]);
      if (j2>=0) upd(tar[i],b*tar[j2]);
    }
  }
}
int main()
{
//  freopen("C.in","r",stdin);
//  freopen(".out","w",stdout);
  int T=read();
  prework();
  while(T--) {
    int n=read(),K=read();
    ll ans=tar[n];
    for(int k=1;;k++) {
      int b=-((k&1)?1:-1) ;
      int j1=n - (3*k*k+k)/2*K;
      int j2=n - (3*k*k-k)/2*K;
```

```
91          if (j1<0&&j2<0) break;
92          if (j1>=0) upd(ans,b*tar[j1]);
93          if (j2>=0) upd(ans,b*tar[j2]);
94        }
95        printf("%I64d\n",ans);
96      }
97
98
99      return 0;
100   }
```

”

1.46 PSLG

```cpp
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define ALL(a) (a).begin(), (a).end()
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
ll sqr(ll a){return a*a;}
ld sqr(ld a){return a*a;}
double sqr(double a){return a*a;}

const double eps=1e-8;
```

```
int dcmp(double x) {
  if (fabs(x)<eps) return 0; else return x<0 ? -1 : 1;
}
ld PI = 3.14159265358979323846264338328;
class P{
public:
  double x,y;
  P(double x=0,double y=0):x(x),y(y){}
  friend long double dis2(P A,P B){return
   ↪  sqr(A.x-B.x)+sqr(A.y-B.y);  }
  friend long double Dot(P A,P B) {return A.x*B.x+A.y*B.y; }
  friend long double Length(P A) {return sqrt(Dot(A,A)); }
  friend long double Angle(P A,P B) {return acos(Dot(A,B) /
   ↪  Length(A) / Length(B) ); }
  friend P operator- (P A,P B) { return P(A.x-B.x,A.y-B.y); }
  friend P operator+ (P A,P B) { return P(A.x+B.x,A.y+B.y); }
  friend P operator* (P A,double p) { return P(A.x*p,A.y*p); }
  friend P operator/ (P A,double p) { return P(A.x/p,A.y/p); }
  friend bool operator< (const P& a,const P& b) {return
   ↪  dcmp(a.x-b.x)<0 ||(dcmp(a.x-b.x)==0&& dcmp(a.y-b.y)<0 );}

};
bool operator==(const P& a,const P& b) {
  return dcmp(a.x-b.x)==0 && dcmp(a.y-b.y) == 0;
}
typedef P V;

double Cross(V A,V B) {return A.x*B.y - A.y*B.x;}
double Area2(P A,P B,P C) {return Cross(B-A,C-A);}

P GetLineIntersection(P p,V v,P Q,V w){
  V u = p-Q;
  double t = Cross(w,u)/Cross(v,w);
  return p+v*t;
}
P GetLineIntersectionB(P p,V v,P Q,V w){
  return GetLineIntersection(p,v-p,Q,w-Q);
}

bool SegmentProperIntersection(P a1,P a2,P b1,P b2) {
  double  c1 = Cross(a2-a1,b1-a1) , c2 = Cross(a2-a1,b2-a1),
      c3 = Cross(b2-b1,a1-b1) , c4 = Cross(b2-b1,a2-b1);
  return dcmp(c1)*dcmp(c2)<0 && dcmp(c3)*dcmp(c4)<0;
}
bool OnSegment(P p,P a1,P a2) {
  return dcmp(Cross(a1-p,a2-p)) == 0 && dcmp(Dot(a1-p,a2-p))<0;
```

```
}
P read_point() {
  P a;
  scanf("%lf%lf",&a.x,&a.y);
  return a;
}

typedef vector<P> Polygon;
double PolygonArea(Polygon &p) {
  double area=0;
  int n=p.size();
  For(i,n-2) area+=Cross(p[i]-p[0],p[i+1]-p[0]);
  return area/2;
}


struct Edge{
  int from,to;
  double ang;
  Edge(int _from,int _to,double _ang):from(_from),
    to(_to),ang(_ang){}
};
#define MAXN (11111+10)
struct PSLG {
  int n,m,face_cnt;
  ld x[MAXN],y[MAXN];
  vector<Edge> edges;
  vi G[MAXN];
  int vis[MAXN*2],left[MAXN*2],prev[MAXN*2];
  vector<Polygon> faces;
  double area[MAXN];
  void init(int n) {
    this->n=n;
    Rep(i,n) G[i].clear();
    edges.clear();
    faces.clear();
  }
  double getAngle(int from,int to) {
    return atan2(y[to]-y[from],x[to]-x[from]);
  }
  void AddEdge(int from,int to) {
    edges.pb(Edge(from,to,getAngle(from,to)));
    edges.pb(Edge(to,from,getAngle(to,from)));
    m=SI(edges);
    G[from].pb(m-2);
    G[to].pb(m-1);
```

```
  }
  void Build() {
    Rep(u,n) {
      int d=SI(G[u]);
      Rep(i,d) {
        Fork(j,i+1,d-1) {
          if (edges[G[u][i]].ang>edges[G[u][j]].ang) {
            swap(G[u][i],G[u][j]);
          }
        }
      }
      Rep(i,d) {
        prev[G[u][(i+1)%d]]=G[u][i];
      }
    }
    MEM(vis)
    face_cnt=0;
    Rep(u,n) {
      Rep(i,SI(G[u])) {
        int e=G[u][i];
        if (!vis[e]) {
          face_cnt++;
          Polygon poly;
          while(1) {
            vis[e]=1;
            left[e]=face_cnt;
            int from = edges[e].from;
            P p(x[from],y[from]);
            poly.pb(p);
            e=prev[e^1];
            if (e==G[u][i]) break;
          }
          faces.pb(poly);
        }
      }
    }
    Rep(i,face_cnt) {
      area[i]=PolygonArea(faces[i]);
    }
  }
}g;

int n, sz;
P p1[MAXN];
void find_path() {
  vector<P> v;
```

```
180   vector<double> dis[MAXN];
181   Rep(i,n) v.pb(p1[i]);
182   Rep(i,n) {
183     Fork(j,i+1,n-1)
184       if (SegmentProperIntersec-
          ↪  tion(p1[i],p1[(i+1)%n],p1[j],p1[(j+1)%n]))
          ↪  {
185         P
            ↪  p=GetLineIntersectionB(p1[i],p1[(i+1)%n],p1[j],p1[(j+1)%n]);
186         v.pb(p);
187         dis[i].pb(Length(p-p1[i]));
188         dis[j].pb(Length(p-p1[j]));
189       }
190   }
191   sort(ALL(v));
192   v.erase( unique(ALL(v)), v.end() );
193   sz=SI(v);
194
195   g.init(sz);
196   Rep(i,sz) g.x[i]=v[i].x,g.y[i]=v[i].y;
197
198   Rep(i,n) {
199     V v1 = p1[(i+1)%n]-p1[i];
200     double len=Length(v1);
201     dis[i].pb(0); dis[i].pb(len);
202     v1=v1/len;
203     sort(ALL(dis[i]));
204     dis[i].erase(unique(ALL(dis[i])),dis[i].end());
205     int tot=SI(dis[i]);
206     Rep(j,tot-1) {
207       P now=p1[i]+v1*dis[i][j];
208       P now2=p1[i]+v1*dis[i][j+1];
209       int id1=lower_bound(ALL(v),now)-v.begin();
210       int id2=lower_bound(ALL(v),now2)-v.begin();
211       if (id1==id2) continue;
212       g.AddEdge(id1,id2);
213     }
214   }
215   g.Build();
216 }
217 void simplify(Polygon& poly) {
218   Polygon ans;
219   int n=SI(poly);
220   Rep(i,n) {
221     if (dcmp(Cross(poly[i]-poly[(i+1)%n],poly[(i+1)%n]-
        ↪  poly[(i+2)%n]))!=0)
```

```
222         ans.pb(poly[(i+1)%n]);
223       }
224       n=SI(ans);
225       cout<<n<<endl;
226       Rep(i,n) printf("%.4lf %.4lf\n",ans[i].x,ans[i].y);
227     }
228
229     int main()
230     {
231     //  freopen("la3218.in","r",stdin);
232     //  freopen("la3218.out","w",stdout);
233       while(scanf("%d",&n)==1&&n) {
234         Rep(i,n) {
235           p1[i]=read_point();
236         }
237         find_path();
238         Polygon poly;
239         Rep(i,g.face_cnt) if (dcmp(g.area[i])<0) {
240           poly=g.faces[i];
241           reverse(ALL(poly));
242           break;
243         }
244         simplify(poly);
245       }
246       return 0;
247     }
```

”

1.47 rat

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
ll gcd(ll a,ll b){if (!b) return a; return gcd(b,a%b);}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
```

```
  return x*f;
}
ll sqr(ll a){return a*a;}
ld sqr(ld a){return a*a;}
double sqr(double a){return a*a;}

struct Rat{
  ll s,m;
  Rat(ll _s=0,ll _m=1) {
    s=_s,m=_m;
    ll d=gcd(s,m);
    s/=d,m/=d;
    if (m<0) s=-s,m=-m;
  }
  Rat operator+ (const Rat &u) {ll d = gcd(m,u.m); return
   ↪  Rat(u.m/d*s+ m/d*u.s, m/d*u.m);  }
  Rat operator- (const Rat &u) {ll d = gcd(m,u.m); return
   ↪  Rat(u.m/d*s- m/d*u.s, m/d*u.m);  }
  Rat operator* (const Rat &u) {return Rat(s*u.s, m*u.m);  }

    bool operator < (const Rat& u) const { return s*u.m < u.s*m; }
  friend inline int dcmp(Rat u) {return (u.s==0)?0:(u.s>0?1:-1);}
};
int main()
{
//  freopen(".in","r",stdin);
//  freopen(".out","w",stdout);


  return 0;
}
```

”

1.48 SA

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (1000000)
#define Sigma_size (1000)
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
class SA
{
public:
  char s[MAXN];
  int sa[MAXN],t[MAXN],t2[MAXN],c[MAXN],n;
  SA(){}
  SA(char *_s){strcpy(s,_s);n=strlen(s);MEM(sa) MEM(t) MEM(t2)
  ↪  MEM(c) }
  void mem(char *_s){strcpy(s,_s);n=strlen(s);MEM(sa) MEM(t)
  ↪  MEM(t2) MEM(c) }
  void build_sa(int m)
  {
```

```
    int *x=t,*y=t2;
    Rep(i,m) c[i]=0;
    Rep(i,n) c[x[i]=s[i]]++;
    For(i,m-1) c[i]+=c[i-1];
    RepD(i,n-1) sa[--c[x[i]]]=i;
    for(int k=1;k<=n;k<<=1)
    {
      int p=0;
      ForkD(i,n-k,n-1) y[p++]=i;
      Rep(i,n) if (sa[i]>=k) y[p++]=sa[i]-k;

      Rep(i,m) c[i]=0;
      Rep(i,n) c[x[y[i]]]++;
      For(i,m-1) c[i]+=c[i-1];
      RepD(i,n-1) sa[--c[x[y[i]]]]=y[i];
      swap(x,y);
      p=1; x[sa[0]]=0;
      For(i,n-1)
        x[sa[i]]=(y[sa[i-1]]==y[sa[i]]&&y[sa[i]+k]==y[sa[i-1]+k])
        ↪ ? (p-1):(p++);
      if (p>=n) break;
      m=p;
    }
  }
  int rank[MAXN],height[MAXN];
  void make_height()
  {
    int k=0;
    Rep(i,n) rank[sa[i]]=i;
    Rep(i,n)
    {
      if (rank[i]-1<0) continue;
      if (k) k--;
      int j=sa[rank[i]-1];
      while(max(i,j)+k<n&&s[i+k]==s[j+k]) ++k;
      height[rank[i]]=k;
    }
  }
  int m; //模板串 P 的长度要事先赋值
  int cmp_suffix(char *pattern,int p)
  {
     return strncmp(pattern,s+sa[p],m);
  }

  int find(char *P)
```

```
87    {
88      m=strlen(P); //这里赋值也行
89      if (cmp_suffix(P,0)<0||cmp_suffix(P,n-1)>0) return -1;
90      int L=0,R=n-1;
91      while(L<=R)
92      {
93        int M=(L+R)>>1;
94        int res=cmp_suffix(P,M);
95        if (!res) return M;
96        else if (res<0) R=M-1;
97        else L=M+1;
98      }
99      return -1;
100   }
101   #define MAXLog (20)
102   int d[MAXN][MAXLog];
103   void RMQ_init()
104   {
105     Rep(i,n) d[i][0]=height[i];
106     for(int j=1;(1<<j)<=n;j++)
107       for(int i=0;i + (1<<j) -1 < n; i++)
108       {
109         d[i][j]=min(d[i][j-1],d[i+(1<<(j-1))][j-1]);
110       }
111    }
112    int query(int L,int R)
113    {
114      if (L>R) swap(L,R);
115      int k=floor(log(R-L+1)/log(2));
116     return min(d[L][k],d[R-(1<<k)+1][k]);
117    }
118   int lcp(int x,int y) { //retrurn lcp(s[x..n-1],s[y..n-1])
119     x=rank[x],y=rank[y];
120     if (x>y) swap(x,y);
121     return query(x+1,y);
122   }
123 }S;
124 char s[MAXN]="aabaa";
125 int main()
126 {
127 //  freopen(".in","r",stdin);
128 //  freopen(".out","w",stdout);
129
130 //  scanf("%s",s);
131   S.mem(s);
132   S.build_sa(Sigma_size);
```

```
133
134     return 0;
135 }
```

”

1.49 SAM

```cpp
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (90000*2+10)
#define Sigmasize (26)
typedef unsigned long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
class SAM {
public:
  char s[MAXN];
  int n;
  SAM():n(0){MEM(s) MEM(son) MEM(pre) MEM(step) last=tot=0;}
  SAM(char *_s){n=strlen(_s),memcpy(s,_s,sizeof(char)*(n+5));
  ↪  MEM(son) MEM(pre) MEM(step) last=tot=0;}
  void mem(){n=0; MEM(s) MEM(son) MEM(pre) MEM(step) last=tot=0;}
  void mem(char *_s){n=strlen(_s);memcpy(s,_s,sizeof(char)*(n+5));
  ↪  MEM(son) MEM(pre) MEM(step) last=tot=0;  }

  int son[MAXN][Sigmasize+1],pre[MAXN],step[MAXN],last,tot;

  void extend(char ch)
  {
    step[++tot]=step[last]+1;
    int p=last,np=tot;

    for(;!son[p][ch];p=pre[p]) son[p][ch]=np;
    if (!p) pre[np]=1;
    else {
```

```
        int q=son[p][ch];
        if (step[q]==step[p]+1) pre[np]=q;
        else {
          step[++tot]=step[p]+1;
          int nq=tot;
          memcpy(son[nq],son[q],sizeof(son[q]));
          pre[nq]=pre[q];
          pre[q]=pre[np]=nq;
          for(;son[p][ch]==q;p=pre[p]) son[p][ch]=nq;
        }
      }
      last=np;
    }

    void build(){
      last=tot=1;
      Rep(i,n) extend(s[i]-'a');
    }



}S1,S2;
char s1[MAXN],s2[MAXN];
int main()
{
//  freopen(".in","r",stdin);

    int T;cin>>T;
    while(T--) {
      scanf("%s%s",s1,s2);
      S1.mem(s1),S2.mem(s2);
      S1.build();
      S2.build();


    }

    return 0;
}
```

”

1.50 SAM_with_furtherMessage

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
#include<iomanip>
#include<vector>
#include<string>
#include<queue>
#include<stack>
#include<map>
#include<sstream>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (40000+10)
#define Sigmasize (26)
typedef unsigned long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
class SAM {
public:
  char s[MAXN];
  int n;
  SAM():n(0){MEM(s) MEM(son) MEM(pre) MEM(step) last=tot=0;}
```

```
44  SAM(char *_s){n=strlen(_s),memcpy(s,_s,sizeof(char)*(n+5));
    ↪  MEM(son) MEM(pre) MEM(step) last=tot=0;}
45  void mem(){n=0; MEM(s) MEM(son) MEM(pre) MEM(step) last=tot=0;}
46  void mem(char *_s){n=strlen(_s);memcpy(s,_s,sizeof(char)*(n+5));
    ↪  MEM(son) MEM(pre) MEM(step) last=tot=0; MEMI(l) MEMi(r)
    ↪  MEM(c) MEM(q)}
47
48  int son[MAXN][Sigmasize+1],pre[MAXN],step[MAXN],last,tot;
49  int l[MAXN],r[MAXN];
50  void extend(char ch)
51  {
52    step[++tot]=step[last]+1;
53    int p=last,np=tot;
54    l[tot]=r[tot]=step[tot];
55
56    for(;!son[p][ch];p=pre[p]) son[p][ch]=np;
57    if (!p) pre[np]=1;
58    else {
59      int q=son[p][ch];
60      if (step[q]==step[p]+1) pre[np]=q;
61      else {
62        step[++tot]=step[p]+1;
63        int nq=tot;
64        memcpy(son[nq],son[q],sizeof(son[q]));
65        pre[nq]=pre[q];
66        pre[q]=pre[np]=nq;
67        for(;son[p][ch]==q;p=pre[p]) son[p][ch]=nq;
68      }
69    }
70    last=np;
71  }
72
73  void build(){
74    last=tot=1;
75    Rep(i,n) extend(s[i]-'a');
76  }
77  int c[MAXN],q[MAXN],len[MAXN];
78  void calc() {
79    MEM(c)
80    For(i,tot) c[step[i]]++;
81    For(i,tot) c[i]+=c[i-1];
82    For(i,tot) q[c[step[i]]--]=i;
83
84    ForD(i,tot) {
85      int u=q[i];
86      l[pre[u]]=min(l[pre[u]],l[u]);
```

```
      r[pre[u]]=max(r[pre[u]],r[u]);
    }
    MEM(len)
    For(i,tot) {
      len[i]=step[i]-step[pre[i]];
    }

  }
  int dfs(int x,int len) {
    For(i,len) {
      Rep(c,26) if (son[x][c]) {
        x=son[x][c]; break;
      }
    }
    return l[x];
  }


}S1;
char s[MAXN];
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
int main()
{
  freopen("H.in","r",stdin);
  int T=read();
  while(T--) {
    scanf("%s",s);
    int n=strlen(s);
    strncpy(s+n,s,n); s[2*n]=0;
    S1.mem(s);
    S1.build();
    S1.calc();
    printf("%d\n",S1.dfs(1,n)+1-n);
  }
  return 0;
}
```

”

1.51 SegmentTree2

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEM2(a,i) memset(a,i,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (100000)
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
class SegmentTree
{
    ll
    ↪  a[MAXN*4],minv[MAXN*4],sumv[MAXN*4],maxv[MAXN*4],addv[MAXN*4],setv[MAXN*4];
    int n;
public:
    SegmentTree(){MEM(a) MEM(minv) MEM(sumv) MEM(addv)
    ↪  MEM2(setv,-1) }
    SegmentTree(int _n):n(_n){MEM(a) MEM(minv) MEM(sumv) MEM(addv)
    ↪  MEM2(setv,-1) }
    void mem(int _n)
    {
        n=_n;
```

```
42          MEMI(a) MEMI(minv)
43      }
44
45      void maintain(int o,int L,int R)
46      {
47      sumv[o]=maxv[o]=minv[o]=0;
48        if (L<R) //只考虑左右子树
49      {
50        sumv[o]=sumv[Lson]+sumv[Rson];
51        minv[o]=min(minv[Lson],minv[Rson]);
52        maxv[o]=max(maxv[Lson],maxv[Rson]);
53      } //只考虑 add 操作
54      minv[o]+=addv[o];maxv[o]+=addv[o];sumv[o]+=addv[o]*(R-L+1);
55      }
56
57    int y1,y2,v;
58    void update(int o,int L,int R)
59    {
60      if (y1<=L&&R<=y2) {
61        addv[o]+=v;
62      }
63      else{
64        int M=(R+L)>>1;
65        if (y1<=M) update(Lson,L,M);
66        if (M< y2) update(Rson,M+1,R);
67      }
68
69      maintain(o,L,R);
70    }
71
72    ll _min,_max,_sum;
73    void query(int o,int L,int R,ll add)
74    {
75      if (y1<=L&&R<=y2)
76      {
77        _sum+=sumv[o]+add*(R-L+1);
78        _min=min(_min,minv[o]+add);
79        _max=max(_max,maxv[o]+add);
80      }
81      else{
82        int M=(R+L)>>1;
83        if (y1<=M) query(Lson,L,M,add+addv[o]);
84        if (M< y2) query(Rson,M+1,R,add+addv[o]);
85      }
86    }
87  }S;
```

```
int main()
{
//  freopen(".in","r",stdin);
//  freopen(".out","w",stdout);
  int n=1;
  S.mem(1);
  S.y1=1,S.y2=1,S.v=1;
  S.update(1,1,n);
  S.y1=1,S.y2=1,S.v=1;
  S.update(1,1,n);
  S._sum=0;
  S.query(1,1,n,0);
  cout<<S._sum;



  return 0;
}
```

”

1.52 SegmentTree2D

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,0x3f,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEMx(a,b) memset(a,b,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
```

```
  return x*f;
}
#define MAXN (2000+10)
struct IntervalTree2D{
  int Max[MAXN][MAXN],Min[MAXN][MAXN],n,m;
  int xo,xleaf,x1,y1,x2,y2,x,y,v,vmax,vmin;
  void query1D(int o,int L,int R) {
    if (y1<=L&&R<=y2) {
      vmax=max(Max[xo][o],vmax);
      vmin=min(Min[xo][o],vmin);
    } else {
      int M=(L+R)>>1;
      if (y1<=M) query1D(Lson,L,M);
      if (M<y2) query1D(Rson,M+1,R);
    }
  }
  void query2D(int o,int L,int R) {
    if (x1<=L&&R<=x2) {
      xo=o; query1D(1,1,m);
    } else {
      int M=(L+R)>>1;
      if (x1<=M) query2D(Lson,L,M);
      if (M<x2) query2D(Rson,M+1,R);
    }
  }
  void modify1D(int o,int L,int R) {
    if (L==R) {
      if (xleaf) { Max[xo][o] = Min[xo][o] = v; return ;}
      Max[xo][o]=max(Max[xo<<1][o],Max[(xo<<1)|1][o]);
      Min[xo][o]=min(Min[xo<<1][o],Min[(xo<<1)|1][o]);
    } else {
      int M=(L+R)>>1;
      if (y<=M) modify1D(Lson,L,M);
      else modify1D(Rson,M+1,R);
      Max[xo][o]=max(Max[xo][Lson],Max[xo][Rson]);
      Min[xo][o]=min(Min[xo][Lson],Min[xo][Rson]);
    }
  }
  void modify2D(int o,int L,int R) {
    if (L==R) {
      xo=o; xleaf=1;modify1D(1,1,m); return;
    }
    int M=(L+R)>>1;
    if (x<=M) modify2D(Lson,L,M);
    else modify2D(Rson,M+1,R);
    xo=o; xleaf=0; modify1D(1,1,m);
```

```
  }
  void query(){vmax=-INF,vmin=INF; query2D(1,1,n);}
}S;
int main()
{
//  freopen("uva11297.in","r",stdin);
//  freopen(".out","w",stdout);
  int n=read(),m=n;
  S.n=n,S.m=m;
  For(i,n) For(j,m) {
    S.v=read();S.x=i,S.y=j;
    S.modify2D(1,1,n);
  }
  int q=read();
  while(q--) {
    char s[10];
    cin>>s;
    if (s[0]=='q') {
      S.x1=read(),S.y1=read(),S.x2=read(),S.y2=read();
      S.query();
      cout<<S.vmax<<' '<<S.vmin<<endl;
    } else {
      S.x=read(),S.y=read(),S.v=read();
      S.modify2D(1,1,n);
    }
  }
  return 0;
}
```

”

1.53 SegmentTree3

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (100000)
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
class SegmentTree
{
    ll
    ↪ a[MAXN*4],minv[MAXN*4],sumv[MAXN*4],maxv[MAXN*4],addv[MAXN*4],setv[MAXN*4];
    int n;
public:
    SegmentTree(){MEM(a) MEM(minv) }
    SegmentTree(int _n):n(_n){MEM(a) MEM(minv) }
    void mem(int _n)
    {
        n=_n;
        MEMI(a) MEMI(minv)
    }

```

```
44      void maintain(int o,int L,int R)
45      {
46        if (L<R) //只考虑左右子树
47      {
48        sumv[o]=sumv[Lson]+sumv[Rson];
49        minv[o]=min(minv[Lson],minv[Rson]);
50        maxv[o]=max(maxv[Lson],maxv[Rson]);
51      } //只考虑 add 操作
52      minv[o]+=addv[o];maxv[o]+=addv[o];sumv[o]+=addv[o]*(R-L+1);
53      }
54
55    int y1,y2,v;
56    void update(int o,int L,int R)
57    {
58      if (y1<=L&&R<=y2) {
59        addv[o]+=v;
60      }
61      else{
62        int M=(R+L)>>1;
63        if (y1<=M) update(Lson,L,M);
64        if (M< y2) update(Rson,M+1,R);
65      }
66
67      maintain(o,L,R);
68    }
69    void update2(int o,int L,int R)
70    {
71      if (y1<=L&&R<=y2) {
72        setv[o]=v;
73      }
74      else{
75        pushdown(o);
76        int M=(R+L)>>1;
77        if (y1<=M) update(Lson,L,M); else maintain(Lson,L,M); //维护
          ↪ pushodown，再次 maintain
78        if (M< y2) update(Rson,M+1,R); else maintain(Rson,M+1,R);
79      }
80
81      maintain(o,L,R);
82    }
83
84    void pushdown(int o)
85    {
86      if (setv[o]>0)
87      {
88        setv[Lson]=setv[Rson]=setv[o];
```

```
89          addv[Lson]=addv[Rson]=0;
90          setv[o]=-1;
91        }
92        if (addv[o])
93        {
94          addv[Lson]+=addv[o];
95          addv[Rson]+=addv[o];
96          addv[o]=0;
97        }
98      }
99
100     void query2(int o,int L,int R,ll add)
101     {
102       if (setv[o]>=0)
103       {
104         _sum+=setv[o]*(min(R,y2)-max(L,y1)+1);
105         _min=min(_min,setv[o]);
106         _max=max(_max,setv[o]);
107       } else if (y1<=L&&R<=y2)
108       {
109         _sum+=sumv[o];
110         _min=min(_min,minv[o]);
111         _max=max(_max,maxv[o]);
112       } else {
113         int M=(L+R)>>1;
114         if (y1<=M) query2(Lson,L,M,add+addv[o]);
115         if (M< y2) query2(Rson,M+1,R,add+addv[o]);
116
117       }
118     }
119
120     ll _min,_max,_sum;
121     void query(int o,int L,int R,ll add)
122     {
123       if (y1<=L&&R<=y2)
124       {
125         _sum+=sumv[o]+add*(R-L+1);
126         _min=min(_min,minv[o]+add);
127         _max=max(_max,maxv[o]+add);
128       }
129       else{
130         int M=(R+L)>>1;
131         if (y1<=M) query(Lson,L,M,add+addv[o]);
132         if (M< y2) query(Rson,M+1,R,add+addv[o]);
133       }
134     }
```

```
135
136
137
138
139
140         //先 set 后 add
141     }S;
142     int main()
143     {
144       freopen(".in","r",stdin);
145       freopen(".out","w",stdout);
146
147
148
149       return 0;
150     }
```

”

1.54 SegmentTree4

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEM2(a,i) memset(a,i,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (500000+10)
typedef long long ll;

class SegmentTree
{
    ll
    ↪ a[MAXN*4],minv[MAXN*4],sumv[MAXN*4],maxv[MAXN*4],addv[MAXN*4],setv[MAXN*4];
    int n;
public:
    SegmentTree(){MEM(a) MEM(minv) MEM(sumv) MEM(maxv) MEM(addv)
    ↪ MEM2(setv,-1) }
    SegmentTree(int _n):n(_n){MEM(a) MEM(minv) MEM(sumv) MEM(maxv)
    ↪ MEM(addv) MEM2(setv,-1) }
    void mem(int _n)
    {
        n=_n;
        MEM(a) MEM(minv)  MEM(sumv) MEM(maxv) MEM(addv)
        ↪ MEM2(setv,-1)
    }
```

```
41
42      void maintain(int o,int L,int R)
43      {
44
45      sumv[o]=maxv[o]=minv[o]=0;
46        if (L<R) //只考虑左右子树
47      {
48        sumv[o]=sumv[Lson]+sumv[Rson];
49        minv[o]=min(minv[Lson],minv[Rson]);
50        maxv[o]=max(maxv[Lson],maxv[Rson]);
51      } //只考虑 add 操作
52      if (setv[o]>=0)
        ↪  sumv[o]=setv[o]*(R-L+1),minv[o]=maxv[o]=setv[o];
53
54      minv[o]+=addv[o];maxv[o]+=addv[o];sumv[o]+=addv[o]*(R-L+1);
55      }
56
57    int y1,y2,v;
58    void update(int o,int L,int R) //y1,y2,v
59    {
60      if (y1<=L&&R<=y2) {
61        addv[o]+=v;
62      }
63      else{
64        pushdown(o);
65        int M=(R+L)>>1;
66        if (y1<=M) update(Lson,L,M); else maintain(Lson,L,M);
67        if (M< y2) update(Rson,M+1,R); else maintain(Rson,M+1,R);
68      }
69
70      maintain(o,L,R);
71
72    }
73    void update2(int o,int L,int R)
74    {
75      if (y1<=L&&R<=y2) {
76        setv[o]=v;addv[o]=0;
77      }
78      else{
79        pushdown(o);
80        int M=(R+L)>>1;
81        if (y1<=M) update2(Lson,L,M); else maintain(Lson,L,M); //维
          ↪  护 pushodown，再次 maintain
82        if (M< y2) update2(Rson,M+1,R); else maintain(Rson,M+1,R);
83      }
84
```

```
85      maintain(o,L,R);
86    }
87
88    void pushdown(int o)
89    {
90      if (setv[o]>=0)
91      {
92        setv[Lson]=setv[Rson]=setv[o];
93        addv[Lson]=addv[Rson]=0;
94        setv[o]=-1;
95      }
96      if (addv[o])
97      {
98        addv[Lson]+=addv[o];
99        addv[Rson]+=addv[o];
100       addv[o]=0;
101     }
102   }
103   ll _min,_max,_sum;
104
105   void query2(int o,int L,int R,ll add)
106   {
107     if (setv[o]>=0)
108     {
109       _sum+=(setv[o]+addv[o]+add)*(min(R,y2)-max(L,y1)+1);
110       _min=min(_min,setv[o]+addv[o]+add);
111       _max=max(_max,setv[o]+addv[o]+add);
112     } else if (y1<=L&&R<=y2)
113     {
114       _sum+=sumv[o]+add*(R-L+1);
115       _min=min(_min,minv[o]+add);
116       _max=max(_max,maxv[o]+add);
117     } else {
118     //  pushdown(o);
119       int M=(L+R)>>1;
120       if (y1<=M) query2(Lson,L,M,add+addv[o]);// else
          ↪  maintain(Lson,L,M);
121       if (M< y2) query2(Rson,M+1,R,add+addv[o]);// else
          ↪  maintain(Rson,M+1,R);
122     }
123     //maintain(o,L,R);
124   }
125
126   void query(int o,int L,int R,ll add) //y1,y2
127   {
128     if (y1<=L&&R<=y2)
```

```
129       {
130         _sum+=sumv[o]+add*(R-L+1);
131         _min=min(_min,minv[o]+add);
132         _max=max(_max,maxv[o]+add);
133       }
134       else{
135         int M=(R+L)>>1;
136         if (y1<=M) query(Lson,L,M,add+addv[o]);
137         if (M< y2) query(Rson,M+1,R,add+addv[o]);
138       }
139     }
140
141     void add(int l,int r,ll v)
142     {
143       y1=l,y2=r;this->v=v;
144       update(1,1,n);
145     }
146     void set(int l,int r,ll v)
147     {
148       y1=l,y2=r;this->v=v;
149       update2(1,1,n);
150     }
151     ll ask(int l,int r,int b=0)
152     {
153       _sum=0,_min=INF,_max=-1;
154       y1=l,y2=r;
155       query2(1,1,n,0);
156     //  cout<<_sum<<' '<<_max<<' '<<_min<<endl;
157
158       switch(b)
159       {
160         case 1:return _sum;
161         case 2:return _min;
162         case 3:return _max;
163         default:break;
164       }
165     }
166     void print()
167     {
168       For(i,n)
169         cout<<ask(i,i,1)<<' ';
170       cout<<endl;
171
172     }
173
174       //先 set 后 add
```

```
}S;
int main()
{
//  freopen(".in","r",stdin);
//  freopen(".out","w",stdout);
  int n=10;
  S.mem(2);

  S.add(1,2,100);
  S.set(1,1,1);
  S.print();
  S.add(1,1,3);
  S.print();



  return 0;
}
```

”

1.55 segmentTree_qujiankaigen_qujianjia

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,0x3f,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEMx(a,b) memset(a,b,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
#define ALL(x) (x).begin(),(x).end()
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
```

```
45    while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
46    return x*f;
47  }
48
49  #define MAXN (100000+10)
50  ll mark[MAXN<<2],sum[MAXN<<2],mx[MAXN<<2],mn[MAXN<<2];
51  void pushUp(int o) {
52    sum[o]=sum[Lson] + sum[Rson];
53    mx[o]=max(mx[Lson],mx[Rson]);
54    mn[o]=min(mn[Lson],mn[Rson]);
55  }
56  void pushDown(int o,ll m) {
57    if (mark[o]) {
58      sum[Lson]+=(m-(m>>1))*mark[o];
59      sum[Rson]+=(m>>1)* mark[o];
60      mx[Lson]+=mark[o],mx[Rson]+=mark[o];
61      mn[Lson]+=mark[o],mn[Rson]+=mark[o];
62      mark[Lson]+=mark[o],mark[Rson]+=mark[o];
63      mark[o]=0;
64    }
65  }
66  void build(int l,int r,int o) {
67    mark[o]=0;
68    if (l==r) {
69      sum[o]=mx[o]=mn[o]=read();
70      return;
71    }
72    int m=(l+r)>>1;
73    build(l,m,Lson);
74    build(m+1,r,Rson);
75    pushUp(o);
76  }
77  void update(int l,int r,int o,int L,int R,ll c) {
78    if (L<=l&&r<=R) {
79      sum[o]+=c*(r-l+1);
80      mx[o]+=c,mn[o]+=c;
81      mark[o]+=c;
82      return;
83    }
84    pushDown(o,r-l+1);
85    int m=(l+r)>>1;
86    if (L<=m) update(l,m,Lson,L,R,c);
87    if (m<R) update(m+1,r,Rson,L,R,c);
88    pushUp(o);
89  }
90  ll query(int l,int r,int o,int L,int R) {
```

```
91    if (L<=l && r<=R) {
92      return sum[o];
93    }
94    pushDown(o,r-l+1);
95    int m=(l+r)>>1;
96    ll ret=0;
97    if (L<=m) ret+=query(l,m,Lson,L,R);
98    if (m<R) ret+=query(m+1,r,Rson,L,R);
99    return ret;
100 }
101 void gen(int l,int r,int o,int L,int R) {
102   if (L<=l && r<=R) {
103     if ((mx[o]-
        ↪  mn[o]==1&&(int)sqrt(mx[o])!=(int)sqrt(mn[o]))||mx[o]==mn[o])
        ↪  {
104       ll c=(ll)sqrt(mx[o])-mx[o];
105       sum[o]+=c*(r-l+1);
106       mx[o]+=c,mn[o]+=c;
107       mark[o]+=c;
108       return;
109     }
110   }
111   pushDown(o,r-l+1);
112   int m=(l+r)>>1;
113   if (L<=m) gen(l,m,Lson,L,R);
114   if (m<R) gen(m+1,r,Rson,L,R);
115   pushUp(o);
116 }
117
118 int main()
119 {
120 //  freopen("uoj228.in","r",stdin);
121 //  freopen(".out","w",stdout);
122   int n=read(),m=read();
123   build(1,n,1);
124   For(i,m) {
125     int op=read(),x=read(),y=read();
126     switch(op) {
127       case 1:update(1,n,1,x,y,read());break;
128       case 2:gen(1,n,1,x,y);break;
129       case 3:printf("%lld\n",query(1,n,1,x,y));
130     }
131   }
132   return 0;
133 }
```

”

1.56 segmentTree_qujianxiugai_qujianLsumRsumSsum

```cpp
#include<cstdio>
#include<cctype>
#include<iostream>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,0x3f,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEMx(a,b) memset(a,b,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
```

```
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
#define MAXN (500000+10)
ll cover[MAXN<<2],lsum[MAXN<<2],rsum[MAXN<<2],msum[MAXN<<2];
void pushUp(int o,int m) {
  lsum[o]=lsum[Lson];
  rsum[o]=rsum[Rson];
  if (lsum[o]==m-(m>>1)) lsum[o]+=lsum[Rson];
  if (rsum[o]==(m>>1)) rsum[o]+=rsum[Lson];
  msum[o]=max(msum[Lson], max(msum[Rson], rsum[Lson]+lsum[Rson]));
}
void pushDown(int o,ll m) {
  if (cover[o]!=-1) {
    cover[Lson]=cover[Rson]=cover[o];
    msum[Lson]=lsum[Lson]=rsum[Lson]=(m-(m>>1))*cover[o];
    msum[Rson]=lsum[Rson]=rsum[Rson]=(m>>1)*cover[o];
    cover[o]=-1;
  }
}
void build(int l,int r,int o) {
  cover[o]=-1;
  msum[o]=lsum[o]=rsum[o]=r-l+1;
  if (l==r) {
    return;
  }
  int m=(l+r)>>1;
  build(l,m,Lson);
  build(m+1,r,Rson);
  pushUp(o,r-l+1);
}
void update(int l,int r,int o,int L,int R,int c) {
  if (L<=l&&r<=R) {
    cover[o]=c;
    msum[o]=lsum[o]=rsum[o]=(r-l+1)*c;
    return;
  }
  pushDown(o,r-l+1);
  int m=(l+r)>>1;
  if (L<=m) update(l,m,Lson,L,R,c);
  if (m<R) update(m+1,r,Rson,L,R,c);
  pushUp(o,r-l+1);
}
int query(int l,int r,int o,int w) {
  if (l==r) return l;
```

```
  pushDown(o,r-l+1);
  int m=(l+r)>>1;
  if (msum[Lson]>=w) return query(l,m,Lson,w);
  else if (rsum[Lson]+lsum[Rson]>=w) return m-rsum[Lson]+1;
  return query(m+1,r,Rson,w);
}
int main()
{
//  freopen("poj3667.in","r",stdin);
//  freopen(".out","w",stdout);
  int n=read(),m=read();
  build(1,n,1);
  while(m--) {
    int op; int a,b;
    scanf("%d%d",&op,&a);
    if (op==1) {
      if (msum[1]<a) puts("0");
      else {
        int p=query(1,n,1,a);
        cout<<p<<endl;
        update(1,n,1,p,p+a-1,0);
      }
    } else {
      int b=read();
      update(1,n,1,a,a+b-1,1);
    }
  }
  return 0;
}
```

”

1.57 SegmentTree_tao_Treap

```cpp
#include <iostream>
#include <cmath>
#include <algorithm>
#include <cstdio>
#include <cstring>
#include <string>
#include <vector>
#include <map>
#include <functional>
#include <cstdlib>
#include <queue>
#include <stack>
#include <set>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (20161119)
#define ALL(x) (x).begin(),(x).end()
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pri(a,n) for(int i=1;i<n;i++) cout<<a[i]<<'
 ↪  ';cout<<a[n]<<endl;
typedef long long ll;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
```

```
44  void upd(ll &a,ll b){a=(a%F+b%F)%F;}
45  int read()
46  {
47      int x=0,f=1; char ch=getchar();
48      while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
49      while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
50      return x*f;
51  }
52  #define MAXN (3023456)
53  int n,m;
54  struct treap{
55      ll rnd[MAXN],v[MAXN],w[MAXN];
56      int size[MAXN],l[MAXN],r[MAXN],cnt;
57      void mem() {
58          MEM(size) MEM(rnd) MEM(l) MEM(r) MEM(v) MEM(w)
59          cnt=0;
60      }
61      void update(int x)
62      {
63          size[x]=size[l[x]]+size[r[x]]+w[x];
64      }
65      void rturn(int &k)
66      {
67          int t=l[k];l[k]=r[t];r[t]=k;update(k);update(t);k=t;
68      }
69      void lturn(int &k)
70      {
71          int t=r[k];r[k]=l[t];l[t]=k;update(k);update(t);k=t;
72      }
73      void insert(int &x,ll rank)
74      {
75          if(!x)
76          {
77              x=++cnt;
78              v[x]=rank; l[x]=r[x]=0;
79              rnd[x]=rand();size[x]=w[x]=1;
80              return ;
81          }
82          size[x]++;
83          if(v[x]<rank)
84          {
85              insert(r[x],rank);
86              if(rnd[r[x]]<rnd[x])lturn(x);
87          }
88          else if (v[x]>rank)
89          {
```

```
            insert(l[x],rank);
            if(rnd[l[x]]<rnd[x]) rturn(x);
        }else w[x]++ ;
    }
    void del(int &x,ll val) {
        if (!x) return ;
        if (v[x]==val) {
            if (w[x]>1) {--w[x]; --size[x]; return; }
            if (!l[x]||!r[x]) x=l[x]+r[x];
            else if (rnd[l[x]]<rnd[r[x]]) rturn(x),del(x,val);
            else lturn(x),del(x,val);
        }
        else {
            --size[x];
            if (val<v[x]) del(l[x],val); else del(r[x],val);
        }
    }
    // return the pointer
    int lower_bound(int x,ll rank) {
        int ans=-1;
        if (!x) return ans;
        if (v[x]<=rank) {
            ans=lower_bound(r[x],rank);
            if (ans==-1) ans=x;
        } else ans=lower_bound(l[x],rank);
        return ans;
    }
    int upper_bound(int x,ll rank) {
        int ans=-1;
        if (!x) return ans;
        if (v[x]>rank) {
            ans=upper_bound(l[x],rank);
            if (ans==-1) ans=x;
        } else ans=upper_bound(r[x],rank);
        return ans;
    }
    void pri(int x){
        if (l[x]) pri(l[x]);
        cout<<v[x]<<' ';
        if (r[x]) pri(r[x]);
    }
    int get_rank(int x,ll val) {
        if (!x) return 0;
        if (v[x]==val) return size[l[x]]+1;
        else if (val<v[x]) return get_rank(l[x],val);
        else return get_rank(r[x],val)+size[l[x]]+w[x];
```

```
136         }
137         int how_many_number_lower_than_x(int x,ll val) {
138             if (!x) return 0;
139             if (v[x]==val) return size[l[x]];
140             else if (val<v[x]) return
                ↪  how_many_number_lower_than_x(l[x],val);
141             else return
                ↪  how_many_number_lower_than_x(r[x],val)+size[l[x]]+w[x];
142         }
143         int get_kth(int x,int k) {
144             if (!x) return 0;
145             if (k<=size[l[x]]) return get_kth(l[x],k);
146             else if (k<=size[l[x]]+w[x]) return x;
147             else return get_kth(r[x],k-size[l[x]]-w[x]);
148         }
149     }T;
150
151     const int maxn =60000;
152     int root[maxn<<2];
153     ll a[maxn];
154     void build(int l,int r,int o) {
155       Fork(i,l,r) T.insert(root[o],a[i]);
156         if (l==r) {
157           return ;
158       }
159       int m=(l+r)>>1;
160       build(l,m,Lson),build(m+1,r,Rson);
161     }
162     void update(int l,int r,int o,int p,ll v) {
163       T.del(root[o],a[p]);
164         T.insert(root[o],v);
165         if (l==r) {
166         return;
167       }
168       int m=(l+r)>>1;
169       if (p<=m) update(l,m,Lson,p,v);
170       else update(m+1,r,Rson,p,v);
171     }
172     ll tmp;
173     void query_lower_bound(int l,int r,int o,int L,int R,ll v) {
174       if(L<=l && r<=R ) {int p=T.lower_bound(root[o],v); if (p!=-1)
          ↪  tmp=max(tmp,T.v[p]); return;}
175       int m=(l+r)>>1;
176       if(L<=m) query_lower_bound(l,m,Lson,L,R,v);
177       if(m<R) query_lower_bound(m+1,r,Rson,L,R,v);
178     }
```

```
179 void query_upper_bound(int l,int r,int o,int L,int R,ll v) {
180   if(L<=l && r<=R ) {int p=T.upper_bound(root[o],v); if (p!=-1)
      ↪  tmp=min(tmp,T.v[p]); return;}
181   int m=(l+r)>>1;
182   if(L<=m) query_upper_bound(l,m,Lson,L,R,v);
183   if(m<R) query_upper_bound(m+1,r,Rson,L,R,v);
184 }
185 void query_rank(int l,int r,int o,int L,int R,ll v) {
186   if(L<=l && r<=R )
      ↪  {tmp+=T.how_many_number_lower_than_x(root[o],v); return;}
187   int m=(l+r)>>1;
188   if(L<=m) query_rank(l,m,Lson,L,R,v);
189   if(m<R) query_rank(m+1,r,Rson,L,R,v);
190 }
191 int query_kth(int L,int R,ll v) {
192   int l=0,r=INF,ans;
193     while(l<=r) {
194         int m=(l+r)/2;
195         tmp=0;query_rank(1,n,1,L,R,m);
196         if (tmp<v) l=m+1,ans=m;else r=m-1;
197     }
198     return ans;
199 }
200 int main() {
201     // freopen("bzoj3196.in","r",stdin);
202     n=read();   m=read();
203     For(i,n) a[i]=read();
204     build(1,n,1);
205     For(i,m) {
206         int opt=read(),x=read(),y=read(); ll v;
207         if (opt^3) v=read();
208             // cout<<opt<<' '<<x<<' '<<y<<' '<<v<<endl;
209         switch (opt) {
210             case
                ↪  1:tmp=1;query_rank(1,n,1,x,y,v);printf("%lld\n",tmp);break;
211             case 2:tmp=1;printf("%d\n",query_kth(x,y,v));break;
212             case 3:update(1,n,1,x,y);a[x]=y;break;
213             case 4:tmp=-INF;query_lower_bound(1,n,1,x,y,v-
                ↪  1);printf("%lld\n",tmp);break;
214             case
                ↪  5:tmp=INF;query_upper_bound(1,n,1,x,y,v);printf("%lld\n",tmp);break;
215         }
216     }
217      return 0;
218 }
```

”

1.58 shift-and_Algorithm_CF_754E_Dasha_and_cyclic_table

```cpp
#include<bits/stdc++.h>
#include<iostream>
#include<cstring>
#include<cstdlib>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %I64d\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
#define ALL(x) (x).begin(),(x).end()

#define MAXN (803)
class shift_and{
public:
  int n,r;
  bitset<MAXN> v,f[26],c;
  int s[MAXN],s2[MAXN];
  void prework() {
    //s s2 n r
  }
```

```
45    bool b[MAXN];
46    void work() {
47      Rep(i,26) f[i].reset();  v.reset(); c.reset();
48      Rep(i,r) if (0<=s2[i]&&s2[i]<26) f[s2[i]][i+1]=1;
49      else c[i+1]=1;
50      Rep(i,26) f[i]|=c;
51      v[0]=1;
52      Rep(i,n) {
53        v=v<<1&f[s[i]];
54        v[0]=1;
55        b[i]=v[r];
56      }
57    }
58  }S;
59
60  char A[MAXN][MAXN],B[MAXN][MAXN];
61  int n,m,r,c;
62
63  bool b[1000][1000];
64
65  int main()
66  {
67  //  freopen("E.in","r",stdin);
68    cin>>n>>m;
69    Rep(i,n) cin>>A[i];
70
71    cin>>r>>c;
72    Rep(i,r) cin>>B[i];
73
74    Rep(i,n+r) Rep(j,c+m)
75      A[i][j]=A[i%n][j%m];
76
77
78    memset(b,-1,sizeof(b));
79
80    S.n=c+m,S.r=c;
81    Rep(k,r) Rep(i,n)  {
82      Rep(j,c+m) S.s[j]=A[i][j]-'a';
83      Rep(j,c) S.s2[j]=B[k][j]-'a';
84      S.work();
85      Fork(l,c-1,c+m-2) if (!S.b[l]) {
86        b[((i-k)%n+n)%n][((l-c+1)%m+m)%m]=0;
87      }
88    }
89    Rep(i,n) {
90      Rep(j,m) {
```

```
91          putchar(b[i][j]?'1':'0');
92        }puts("");
93      }
94
95      return 0;
96    }
```

”

1.59 slope_dp

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,0x3f,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEMx(a,b) memset(a,b,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (100000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
```

```
45    return x*f;
46  }
47  int n;
48  pair<ll,ll> p2[100000];
49  ll f[100000];
50  int que[100000]={0};
51  int H = 0 , T = 0;
52  ll getdp(int i,int j) {
53    return f[que[H]] + p2[que[H]+1].fi * p2[i].se;
54  }
55  ll getUp(int j,int k) { //yj-yk
56    return f[j]-f[k];
57  }
58  ll getDown(int j,int k) { // xj-xk
59    return p2[j+1].fi - p2[k+1].fi;
60  }
61  // fi = min( fj + cost(j+1,i) )
62  void slope_dp(int n) {
63    MEM(que) H=T=0;
64    f[0]=0;
65    For(i,n) {
66      while( H<T && getUp(que[H+1],que[H] ) <= getDown(que[H+1],
       ↪  que[H]) * (-p2[i].se ) )    // 条件 que[H+1] 比 que[H] 优
67        ++H;
68      f[i]=getdp(i,que[H]);
69      while( H<T && getUp(i,que[T] ) * getDown(que[T],que[T-1]) >=
       ↪  getUp(que[T], que[T-1] ) * getDown(i, que[T] ) ) // 大等于
       ↪  和下等于取决于  min,max
70        --T;
71      que[++T] =i;
72    }
73  }
74  int main()
75  {
76  //  freopen("bzoj1597.in","r",stdin);
77  //  freopen(".out","w",stdout);
78
79
80    return 0;
81  }
```

”

1.60 SPFA

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (100000+10)
#define MAXM (60000*2+10)
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}

class SPFA
{
public:
  void mem()
  {
    MEM(pre) MEM(edge) MEM(pre) MEM(weight) size=1;
  }
  int q[MAXN*100];
  int edge[MAXM],next[MAXM],pre[MAXN],weight[MAXM],size;
  void addedge(int u,int v,int w)
  {
    edge[++size]=v;
```

```
45      weight[size]=w;
46      next[size]=pre[u];
47      pre[u]=size;
48    }
49    void addedge2(int u,int v,int w){addedge(u,v,w);addedge(v,u,w);}
50    int d[MAXN];
51    bool b[MAXN];
52    int spfa(int s,int t)
53    {
54      MEM(b) MEMI(d)
55      b[s]=1; d[s]=0;
56
57      int head=1,tail=1;q[1]=1;
58      while(head<=tail)
59      {
60        int now=q[head++];
61        b[now]=0;
62        Forp(now)
63        {
64          int v=edge[p];
65          if (d[now]+weight[p]<d[v]) {
66            d[v]=d[now]+weight[p];
67            if (!b[v]) { b[v]=1,q[++tail]=v;
68            }
69          }
70        }
71      }
72      return d[t];
73    }
74  }S1,S2;
75  class link_table
76  {
77  public:
78    void mem()
79    {
80      MEM(pre) MEM(edge) MEM(next) MEM(weight) size=1;
81    }
82    int edge[MAXM],next[MAXM],pre[MAXN],weight[MAXM],size;
83
84    void addedge(int u,int v,int w)
85    {
86      edge[++size]=v;
87      weight[size]=w;
88      next[size]=pre[u];
89      pre[u]=size;
90    }
```

```
91    void addedge2(int u,int v,int w){addedge(u,v,w);addedge(v,u,w);}
92  }St;
93  int n,m;
94  int main()
95  {
96  //  freopen(".in","r",stdin);
97  //  freopen(".out","w",stdout);
98
99    return 0;
100 }
```

”

1.61 Splay

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (300000+10)
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}

int n,m;

class Splay
{
public:
  int father[MAXN],siz[MAXN],n;
  int ch[MAXN][2],val[MAXN];
  bool root[MAXN],rev[MAXN];
  int roo; //root
  void mem(int _n)
  {
    MEM(father) MEM(siz) MEM(root) MEM(rev)  MEM(ch) MEM(val)
    ↪  flag=0;
```

```
44      n=0;
45      roo=1;
46      build(roo,1,_n,0);root[1]=1;
47    }
48    void newnode(int &x,int f,int v)
49    {
50      x=++n;
51      father[x]=f;
52      val[x]=v-1;
53    }
54
55    void build(int &x,int L,int R,int f)
56    {
57      if (L>R) return ;
58      int m=(L+R)>>1;
59      newnode(x,f,m);
60      build(ch[x][0],L,m-1,x);
61      build(ch[x][1],m+1,R,x);
62      maintain(x);
63    }
64    int getkth(int x,int k)
65    {
66      pushdown(x);
67      int t;
68      if (ch[x][0]) t=siz[ch[x][0]]; else t=0;
69
70      if (t==k-1) return x;
71      else if (t>=k) return getkth(ch[x][0],k);
72      else return getkth(ch[x][1],k-t-1);
73
74    }
75
76
77    void pushdown(int x)
78    {
79      if (x) if (rev[x])
80      {
81        swap(ch[x][0],ch[x][1]);
82        if (ch[x][0]) rev[ ch[x][0] ]^=1;
83        if (ch[x][1]) rev[ ch[x][1] ]^=1;
84        rev[x]^=1;
85      }
86    }
87    void maintain(int x)
88    {
89      siz[x]=siz[ch[x][0]]+siz[ch[x][1]]+1;
```

```
90    }
91    void rotate(int x)
92    {
93      int y=father[x],kind=ch[y][1]==x;
94
95      pushdown(y); pushdown(x);
96
97      ch[y][kind]=ch[x][!kind];
98      if (ch[y][kind]) {
99        father[ch[y][kind]]=y;
100     }
101     father[x]=father[y];
102     father[y]=x;
103     ch[x][!kind]=y;
104     if (root[y])
105     {
106       root[x]=1;root[y]=0;roo=x;
107     }
108     else
109     {
110       ch[father[x]][ ch[father[x]][1]==y ] = x;
111     }
112     maintain(y);maintain(x);
113   }
114   void splay(int x)
115   {
116     while(!root[x])
117     {
118       int y=father[x];
119       int z=father[y];
120       if (root[y]) rotate(x);
121       else if ( (ch[y][1]==x)^(ch[z][1]==y) )
122       {
123         rotate(x); rotate(x);
124       }
125       else
126       {
127         rotate(y); rotate(x);
128       }
129     }
130     roo=x;
131   }
132   void splay(int x,int r)
133   {
134     while(!(father[x]==r))
135     {
```

```
136         int y=father[x];
137         int z=father[y];
138         if (father[y]==r) rotate(x);
139         else if ( (ch[y][1]==x)^(ch[z][1]==y) )
140         {
141           rotate(x); rotate(x);
142         }
143         else
144         {
145           rotate(y); rotate(x);
146         }
147       }
148     }
149
150     void Cut(int a,int b,int c)
151     {
152       int x=getkth(roo,a),y=getkth(roo,b);
153       splay(x);
154       splay(y,roo);
155       pushdown(x);pushdown(y);
156       int z=ch[y][0];
157       ch[y][0]=0; maintain(y); maintain(x);
158
159       int u=getkth(roo,c),v=getkth(roo,c+1);
160       splay(u);
161       splay(v,roo);
162       pushdown(u);pushdown(v);
163       ch[v][0]=z;father[z]=v;
164       maintain(v);maintain(u);
165
166     }
167
168     void Flip(int a,int b)
169     {
170       int x=getkth(roo,a),y=getkth(roo,b);
171       splay(x);
172       splay(y,roo);
173       pushdown(x);pushdown(y);
174       int z=ch[y][0];
175       rev[z]^=1;
176       maintain(y); maintain(x);
177     }
178
179     bool flag;
180     void print(int x)
181     {
```

```
182       if (x==0) return ;
183       pushdown(x);
184       print(ch[x][0]);
185 
186       if (val[x]!=0&&val[x]!=n-1)
187       {
188         if (flag) putchar(' '); else flag=1;
189         printf("%d",val[x]);
190 
191       }
192       print(ch[x][1]);
193     }
194 
195 }S;
196 int main()
197 {
198 //  freopen(".in","r",stdin);
199 //  freopen(".out","w",stdout);
200 
201   while(cin>>n>>m)
202   {
203     S.mem(n);
204 
205     S.print(S.roo);cout<<endl;
206 
207   }
208 
209 
210   return 0;
211 }
```

”

1.62 Splay2

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (200000+10)
typedef long long ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+(a-b)/F*F+F)%F;}
int modF(int a,int b){return (a+a/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}

int n,m;
int a[MAXN];
class Splay
{
public:
  int father[MAXN],siz[MAXN],n;
  int ch[MAXN][2],val[MAXN];
  bool root[MAXN],rev[MAXN];
  int addv[MAXN],minv[MAXN];
  int roo; //root
  void mem(int _n)
  {
```

```
45      MEM(father) MEM(siz) MEM(root) MEM(rev)  MEM(ch) MEM(val)
        ↪  flag=0; MEM(addv) MEM(minv)
46      n=0;
47      roo=1;
48      build(roo,1,_n,0);root[1]=1;
49    }
50    void newnode(int &x,int f,int v)
51    {
52      x=++n;
53      father[x]=f;
54      val[x]=minv[x]=v;siz[x]=1;
55    }
56
57    void build(int &x,int L,int R,int f)
58    {
59      if (L>R) return ;
60      int m=(L+R)>>1;
61      newnode(x,f,a[m]);
62      build(ch[x][0],L,m-1,x);
63      build(ch[x][1],m+1,R,x);
64      maintain(x);
65    }
66    int getkth(int x,int k)
67    {
68      pushdown(x);
69      int t;
70      if (ch[x][0]) t=siz[ch[x][0]]; else t=0;
71
72      if (t==k-1) return x;
73      else if (t>=k) return getkth(ch[x][0],k);
74      else return getkth(ch[x][1],k-t-1);
75
76    }
77
78
79    void pushdown(int x)
80    {
81      if (x) if (rev[x])
82      {
83        swap(ch[x][0],ch[x][1]);
84        if (ch[x][0]) rev[ ch[x][0] ]^=1;
85        if (ch[x][1]) rev[ ch[x][1] ]^=1;
86        rev[x]^=1;
87      }
88      if (addv[x])
89      {
```

```
90        if (ch[x][0]) addv[ ch[x][0] ]+=addv[x],minv[ ch[x][0]
          ↪  ]+=addv[x],val[ ch[x][0] ]+=addv[x];
91        if (ch[x][1]) addv[ ch[x][1] ]+=addv[x],minv[ ch[x][1]
          ↪  ]+=addv[x],val[ ch[x][1] ]+=addv[x];
92        addv[x]=0;
93      }
94    }
95    void maintain(int x)
96    {
97      siz[x]=siz[ch[x][0]]+siz[ch[x][1]]+1;
98      minv[x]=val[x];
99      if (ch[x][0]) minv[ x ]=min(minv[x],minv[ ch[x][0] ] + addv[x]
        ↪  );
100     if (ch[x][1]) minv[ x ]=min(minv[x],minv[ ch[x][1] ] + addv[x]
        ↪  );
101   }
102   void rotate(int x)
103   {
104     int y=father[x],kind=ch[y][1]==x;
105
106     pushdown(y); pushdown(x);
107
108     ch[y][kind]=ch[x][!kind];
109     if (ch[y][kind]) {
110       father[ch[y][kind]]=y;
111     }
112     father[x]=father[y];
113     father[y]=x;
114     ch[x][!kind]=y;
115     if (root[y])
116     {
117       root[x]=1;root[y]=0;roo=x;
118     }
119     else
120     {
121       ch[father[x]][ ch[father[x]][1]==y ] = x;
122     }
123     maintain(y);maintain(x);
124   }
125   void splay(int x)
126   {
127     while(!root[x])
128     {
129       int y=father[x];
130       int z=father[y];
131       if (root[y]) rotate(x);
```

```
132         else if ( (ch[y][1]==x)^(ch[z][1]==y) )
133         {
134           rotate(x); rotate(x);
135         }
136         else
137         {
138           rotate(y); rotate(x);
139         }
140       }
141       roo=x;
142     }
143     void splay(int x,int r)
144     {
145       while(!(father[x]==r))
146       {
147         int y=father[x];
148         int z=father[y];
149         if (father[y]==r) rotate(x);
150         else if ( (ch[y][1]==x)^(ch[z][1]==y) )
151         {
152           rotate(x); rotate(x);
153         }
154         else
155         {
156           rotate(y); rotate(x);
157         }
158       }
159     }
160
161     void Cut(int a,int b,int c)
162     {
163       int x=getkth(roo,a),y=getkth(roo,b);
164       splay(x);
165       splay(y,roo);
166       pushdown(x);pushdown(y);
167       int z=ch[y][0];
168       ch[y][0]=0; maintain(y); maintain(x);
169
170       int u=getkth(roo,c),v=getkth(roo,c+1);
171       splay(u);
172       splay(v,roo);
173       pushdown(u);pushdown(v);
174       ch[v][0]=z;father[z]=v;
175       maintain(v);maintain(u);
176
177     }
```

```
178
179   void Flip(int a,int b)
180   {
181     int x=getkth(roo,a),y=getkth(roo,b);
182     splay(x);
183     splay(y,roo);
184     pushdown(x);pushdown(y);
185     int z=ch[y][0];
186     rev[z]^=1;
187     maintain(y); maintain(x);
188   }
189
190   void Add(int a,int b,int c)
191   {
192     int x=getkth(roo,a),y=getkth(roo,b);
193     splay(x);
194     splay(y,roo);
195     pushdown(x);pushdown(y);
196     int z=ch[y][0];
197     addv[z]+=c; val[z]+=c; minv[z]+=c;
198     maintain(y); maintain(x);
199   }
200
201   int queryMin(int a,int b)
202   {
203     int x=getkth(roo,a),y=getkth(roo,b);
204     splay(x);
205     splay(y,roo);
206     pushdown(x);pushdown(y);
207     int z=ch[y][0];
208     maintain(y); maintain(x);
209     return minv[z];
210   }
211
212   void insert(int a,int P)
213   {
214     int x=getkth(roo,a),y=getkth(roo,a+1);
215     splay(x);
216     splay(y,roo);
217     pushdown(x);pushdown(y);
218     newnode(ch[y][0],y,P);
219     maintain(y); maintain(x);
220   }
221   void Delete(int a,int b)
222   {
223     int x=getkth(roo,a),y=getkth(roo,b);
```

```
224       splay(x);
225       splay(y,roo);
226       pushdown(x);pushdown(y);
227       int z=ch[y][0];
228       ch[y][0]=0; father[z]=0; maintain(y); maintain(x);
229
230     }
231
232     bool flag;
233     void print(int x)
234     {
235       if (x==0) return ;
236       pushdown(x);
237       print(ch[x][0]);
238       printf("%d ",val[x]);
239       print(ch[x][1]);
240     }
241
242   }S;
243
244   int main()
245   {
246   //  freopen(".in","r",stdin);
247   //  freopen(".out","w",stdout);
248
249     while(cin>>n)
250     {
251
252       S.print(S.roo);cout<<endl;
253     }
254
255
256     return 0;
257   }
```

”

1.63 SSC

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
#include<vector>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (10000+10)
#define MAXM (50000+10)
long long mul(long long a,long long b){return (a*b)%F;}
long long add(long long a,long long b){return (a+b)%F;}
long long sub(long long a,long long b){return
 ↪  (a-b+(a-b)/F*F+F)%F;}
typedef long long ll;
class SSC
{
public:
  int n,b[MAXN],num[MAXN];
  vector<int> G[MAXN],rG[MAXN]; //图，反向后的图
  vector<int> vs; //后续遍历顶点列表
  void mem(int _n)
  {
    n=_n; MEM(num)
    For(i,n) G[i].clear(),rG[i].clear();
    vs.clear();
  }
  void addedge(int u,int v)
```

```
  {
    G[u].push_back(v);
    rG[v].push_back(u);
  }
  void dfs(int x)
  {
    b[x]=1;
    Rep(i,G[x].size())
    {
      if (!b[G[x][i]]) dfs(G[x][i]);
    }
    vs.push_back(x);
  }
  void rdfs(int x,int k)
  {
    b[x]=1;num[x]=k;
    Rep(i,rG[x].size())
    {
      if (!b[rG[x][i]]) rdfs(rG[x][i],k);
    }
  }
  int ssc()
  {
    MEM(b)
    For(i,n) if (!b[i]) dfs(i);
    MEM(b) int k=0;
    RepD(i,vs.size()-1) if (!b[vs[i]]) rdfs(vs[i],++k);
    return k;
  }

}S;
int main()
{
//  freopen(".in","r",stdin);
//  freopen(".out","w",stdout);

  return 0;
}
```

”

1.64 Stoer_Wagner

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
#include<iomanip>
#include<vector>
#include<string>
#include<queue>
#include<stack>
#include<map>
#include<sstream>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %I64d\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
```

```
45              }
46  #pragma comment(linker, "/STACK:102400000,102400000")
47  typedef long long ll;
48  typedef long double ld;
49  typedef unsigned long long ull;
50  ll mul(ll a,ll b){return (a*b)%F;}
51  ll add(ll a,ll b){return (a+b)%F;}
52  ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
53  void upd(ll &a,ll b){a=(a%F+b%F)%F;}
54  int read()
55  {
56    int x=0,f=1; char ch=getchar();
57    while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
58    while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
59    return x*f;
60  }
61  #define MAXN (600)
62  #define MAXM (25100)
63  #define MAXL (10)
64  struct edge{
65    int u,v,w;
66    edge(int _u=0,int _v=0,int _w=0):u(_u),v(_v),w(_w){}
67  };
68  int n,m;
69  int a[MAXN][MAXN],dis[MAXN],v[MAXN],vis[MAXN];
70  int stoer_wagner(int n) {
71    int ans=INF;
72    For(i,n) v[i]=i;
73    while(n>1) {
74      int p=0,last=0;
75      For(i,2,n) {
76        dis[v[i]] = a[v[1]][v[i]];
77        if (dis[v[i]]>dis[v[p]]) p=i;
78      }
79      MEM(vis)
80      vis[v[1]]=1;
81      For(i,2,n) {
82        if (i==n) {
83          ans=min(ans,dis[v[p]]);
84          For(j,n) a[v[j]][v[last]] = a[v[last]][v[j]] +=
            ↪  a[v[j]][v[p]];
85          v[p]=v[n--];
86        }
87        vis[v[last=p]]=1; p=-1;
88        For(j,2,n) if (!vis[v[j]]) {
89          dis[v[j]] +=a[v[last]][v[j]];
```

```
          if (p==-1 || dis[v[p]]<dis[v[j]]) p=j;
        }
      }
    }
    return ans;
}
int main()
{
//  freopen("E.in","r",stdin);
//  freopen(".out","w",stdout);
    while(cin>>n>>m) {
      MEM(a)
      Rep(i,m) {
        int u=read(),v=read(),w=read();
        ++u,++v;
        if (u!=v) a[u][v]+=w,a[v][u]+=w;
      }
      cout<<stoer_wagner(n)<<endl;
    }
    return 0;
}
```

”

1.65 Treap

```
#include <iostream>
#include <cmath>
#include <algorithm>
#include <cstdio>
#include <cstring>
#include <string>
#include <vector>
#include <map>
#include <functional>
#include <cstdlib>
#include <queue>
#include <stack>
#include <set>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (20161119)
#define ALL(x) (x).begin(),(x).end()
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pri(a,n) for(int i=1;i<n;i++) cout<<a[i]<<'
 ↪  ';cout<<a[n]<<endl;
typedef long long ll;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
```

```
44  void upd(ll &a,ll b){a=(a%F+b%F)%F;}
45  int read()
46  {
47      int x=0,f=1; char ch=getchar();
48      while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
49      while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
50      return x*f;
51  }
52  #define MAXN (123456)
53  int n;
54  struct treap{
55      ll rnd[MAXN],v[MAXN],w[MAXN];
56      int size[MAXN],l[MAXN],r[MAXN],cnt;
57      void mem() {
58          MEM(size) MEM(rnd) MEM(l) MEM(r) MEM(v) MEM(w)
59          cnt=0;
60      }
61      void update(int x)
62      {
63          size[x]=size[l[x]]+size[r[x]]+w[x];
64      }
65      void rturn(int &k)
66      {
67          int t=l[k];l[k]=r[t];r[t]=k;update(k);update(t);k=t;
68      }
69      void lturn(int &k)
70      {
71          int t=r[k];r[k]=l[t];l[t]=k;update(k);update(t);k=t;
72      }
73      void insert(int &x,ll rank)
74      {
75          if(!x)
76          {
77              x=++cnt;
78              v[x]=rank; l[x]=r[x]=0;
79              rnd[x]=rand();size[x]=w[x]=1;
80              return ;
81          }
82          size[x]++;
83          if(v[x]<rank)
84          {
85              insert(r[x],rank);
86              if(rnd[r[x]]<rnd[x])lturn(x);
87          }
88          else if (v[x]>rank)
89          {
```

```
            insert(l[x],rank);
            if(rnd[l[x]]<rnd[x]) rturn(x);
        }else w[x]++ ;
    }
    void del(int &x,ll val) {
        if (!x) return ;
        if (v[x]==val) {
            if (w[x]>1) {--w[x]; --size[x]; return; }
            if (!l[x]||!r[x]) x=l[x]+r[x];
            else if (rnd[l[x]]<rnd[r[x]]) rturn(x),del(x,val);
            else lturn(x),del(x,val);
        }
        else {
            --size[x];
            if (val<v[x]) del(l[x],val); else del(r[x],val);
        }
    }
    // return the pointer
    int lower_bound(int x,ll rank) {
        int ans=-1;
        if (!x) return ans;
        if (v[x]<=rank) {
            ans=lower_bound(r[x],rank);
            if (ans==-1) ans=x;
        } else ans=lower_bound(l[x],rank);
        return ans;
    }
    int upper_bound(int x,ll rank) {
        int ans=-1;
        if (!x) return ans;
        if (v[x]>rank) {
            ans=upper_bound(l[x],rank);
            if (ans==-1) ans=x;
        } else ans=upper_bound(r[x],rank);
        return ans;
    }
    void pri(int x){
        if (l[x]) pri(l[x]);
        cout<<v[x]<<' ';
        if (r[x]) pri(r[x]);
    }
    int get_rank(int x,ll val) {
        if (!x) return 0;
        if (v[x]==val) return size[l[x]]+1;
        else if (val<v[x]) return get_rank(l[x],val);
        else return get_rank(r[x],val)+size[l[x]]+w[x];
```

```
136         }
137         int get_kth(int x,int k) {
138             if (!x) return 0;
139             if (k<=size[l[x]]) return get_kth(l[x],k);
140             else if (k<=size[l[x]]+w[x]) return x;
141             else return get_kth(r[x],k-size[l[x]]-w[x]);
142         }
143     }T;
144     int main() {
145         // freopen("tyvjP1728.in","r",stdin);
146 
147         n=read();
148         int rot=0;
149         For(i,n) {
150             int opt=read(),x=read();
151             // cout<<opt<<" "<<x<<endl;
152             switch (opt) {
153                 case 1:T.insert(rot,x);break;
154                 case 2:T.del(rot,x);break;
155                 case 3:printf("%d\n",T.get_rank(rot,x));break;
156                 case 4:printf("%lld\n",T.v[T.get_kth(rot,x)]);break;
157                 case 5:printf("%lld\n",T.v[T.lower_bound(rot,x-
                    ↪ 1)]);break;
158                 case
                    ↪ 6:printf("%lld\n",T.v[T.upper_bound(rot,x)]);break;
159             }
160             // T.pri(rot);puts("");
161         }
162          return 0;
163     }
```

”

1.66 Tree_chain_subdivision

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
#include<vector>
#include<iomanip>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (100000+10)
#define MAXM (200000+10)
#define MAXV (1000+10)
#define pb push_back
#define mp make_pair
#pragma comment(linker, "/STACK:1024000000,1024000000")
typedef int ll;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
struct Chain{
  int u,v,w;
  Chain(){}
  Chain(int _u,int _v,int _w):u(_u),v(_v),w(_w){}
};
vector<Chain> a[MAXN];
int n,m;
```

```cpp
int edge[MAXM],Next[MAXM],Pre[MAXN],siz=1;
void addedge(int u,int v)
{
  edge[++siz]=v;
  Next[siz]=Pre[u];
  Pre[u]=siz;
}
void addedge2(int u,int v){addedge(u,v);addedge(v,u);}

bool vis[MAXN];
int cnt,id[MAXN];
int son[MAXN],dep[MAXN],sz[MAXN],top[MAXN],pre[MAXN],q[MAXN];
void build()
{
  MEM(vis) cnt=0; MEM(id)
  MEM(son) MEM(dep) MEM(sz) MEM(top) MEM(pre) MEM(q)
  int r=1;
  vis[dep[1]=q[1]=1]=1;
  For(i,r)
  {
    int u=q[i];
    Forp(u)
    {
      int v=edge[p];
      if (vis[v]) continue; else vis[v]=1;
      dep[ q[++r]=v ]=dep[u]+1;
      pre[v]=u;
    }
  }
  ForD(i,r) {
    sz[pre[q[i]]] += ++sz[q[i]];
    if (sz[son[pre[q[i]]]]<sz[q[i]] ) son[pre[q[i]]] = q[i];
  }
  For(i,r) {
    if (!top[q[i]])
      for(int x=q[i];x;x=son[x]) {
        top[x]=q[i];
        id[x]=++cnt;
      }
  }

}
int lca(int a,int b)
{
  while(1) {
    if (top[a]==top[b]) return dep[a]<=dep[b] ? a:b;
```

```
91      if (dep[top[a]]<dep[top[b]]) swap(a,b);
92      a=pre[top[a]];
93    }
94  }
95
96
97  #define MEM2(a,i) memset(a,i,sizeof(a));
98  class SegmentTree
99  {
100     ll
        ↪  a[MAXN*4],minv[MAXN*4],sumv[MAXN*4],maxv[MAXN*4],addv[MAXN*4],setv[MAXN*4];
101     int n;
102 public:
103     SegmentTree(){MEM(a) MEM(minv) MEM(sumv) MEM(maxv) MEM(addv)
        ↪  MEM2(setv,-1) }
104     SegmentTree(int _n):n(_n){MEM(a) MEM(minv) MEM(sumv) MEM(maxv)
        ↪  MEM(addv) MEM2(setv,-1) }
105     void mem(int _n)
106     {
107         n=_n;
108         MEM(a) MEM(minv)  MEM(sumv) MEM(maxv) MEM(addv)
            ↪  MEM2(setv,-1)
109     }
110
111     void maintain(int o,int L,int R)
112     {
113
114     sumv[o]=maxv[o]=minv[o]=0;
115       if (L<R) //只考虑左右子树
116     {
117       sumv[o]=sumv[Lson]+sumv[Rson];
118       minv[o]=min(minv[Lson],minv[Rson]);
119       maxv[o]=max(maxv[Lson],maxv[Rson]);
120     } //只考虑 add 操作
121     if (setv[o]>=0)
        ↪  sumv[o]=setv[o]*(R-L+1),minv[o]=maxv[o]=setv[o];
122
123     minv[o]+=addv[o];maxv[o]+=addv[o];sumv[o]+=addv[o]*(R-L+1);
124     }
125
126   int y1,y2,v;
127   void update(int o,int L,int R) //y1,y2,v
128   {
129     if (y1<=L&&R<=y2) {
130       addv[o]+=v;
131     }
```

```
132       else{
133         pushdown(o);
134         int M=(R+L)>>1;
135         if (y1<=M) update(Lson,L,M); else maintain(Lson,L,M);
136         if (M< y2) update(Rson,M+1,R); else maintain(Rson,M+1,R);
137       }
138
139       maintain(o,L,R);
140
141     }
142     void update2(int o,int L,int R)
143     {
144       if (y1<=L&&R<=y2) {
145         setv[o]=v;addv[o]=0;
146       }
147       else{
148         pushdown(o);
149         int M=(R+L)>>1;
150         if (y1<=M) update2(Lson,L,M); else maintain(Lson,L,M); //维
            ↪ 护 pushodown，再次 maintain
151         if (M< y2) update2(Rson,M+1,R); else maintain(Rson,M+1,R);
152       }
153
154       maintain(o,L,R);
155     }
156
157     void pushdown(int o)
158     {
159       if (setv[o]>=0)
160       {
161         setv[Lson]=setv[Rson]=setv[o];
162         addv[Lson]=addv[Rson]=0;
163         setv[o]=-1;
164       }
165       if (addv[o])
166       {
167         addv[Lson]+=addv[o];
168         addv[Rson]+=addv[o];
169         addv[o]=0;
170       }
171     }
172     ll _min,_max,_sum;
173
174     void query2(int o,int L,int R,ll add)
175     {
176       if (setv[o]>=0)
```

```
177       {
178         _sum+=(setv[o]+addv[o]+add)*(min(R,y2)-max(L,y1)+1);
179         _min=min(_min,setv[o]+addv[o]+add);
180         _max=max(_max,setv[o]+addv[o]+add);
181       } else if (y1<=L&&R<=y2)
182       {
183         _sum+=sumv[o]+add*(R-L+1);
184         _min=min(_min,minv[o]+add);
185         _max=max(_max,maxv[o]+add);
186       } else {
187       //  pushdown(o);
188         int M=(L+R)>>1;
189         if (y1<=M) query2(Lson,L,M,add+addv[o]);// else
            ↪  maintain(Lson,L,M);
190         if (M< y2) query2(Rson,M+1,R,add+addv[o]);// else
            ↪  maintain(Rson,M+1,R);
191       }
192       //maintain(o,L,R);
193     }
194
195     void query(int o,int L,int R,ll add) //y1,y2
196     {
197       if (y1<=L&&R<=y2)
198       {
199         _sum+=sumv[o]+add*(R-L+1);
200         _min=min(_min,minv[o]+add);
201         _max=max(_max,maxv[o]+add);
202       }
203       else{
204         int M=(R+L)>>1;
205         if (y1<=M) query(Lson,L,M,add+addv[o]);
206         if (M< y2) query(Rson,M+1,R,add+addv[o]);
207       }
208     }
209
210     void add(int l,int r,ll v)
211     {
212       if (l>r) swap(l,r);
213       y1=l,y2=r;this->v=v;
214       update(1,1,n);
215     }
216     void set(int l,int r,ll v)
217     {
218       y1=l,y2=r;this->v=v;
219       update2(1,1,n);
220     }
```

```
221   ll ask(int l,int r,int b=1)
222   {
223     if (l>r) swap(l,r);
224     _sum=0,_min=INF,_max=-1;
225     y1=l,y2=r;
226     query2(1,1,n,0);
227     switch(b)
228     {
229       case 1:return _sum;
230       case 2:return _min;
231       case 3:return _max;
232       default:break;
233     }
234   }
235   void print()
236   {
237     For(i,n)
238       cout<<ask(i,i,1)<<' ';
239     cout<<endl;
240 
241   }
242 
243     //先 set 后 add
244 }S[2];   //sum & dp
245 
246 ll Ask(int a,int b,int f)
247 {
248   ll ans=0;
249   while (top[a]^top[b]) {
250     if (dep[top[a]]<dep[top[b]]) swap(a,b);
251     ans+=S[f].ask(id[top[a]],id[a],1);
252     a=pre[top[a]];
253   }
254   if (dep[a]>dep[b]) swap(a,b);
255   ans+=S[f].ask(id[a],id[b],1);
256   return ans;
257 }
258 
259 
260 int main()
261 {
262 //  freopen("hdu5293.in","r",stdin);
263 
264 
265   return 0;
266 }
```

”

1.67 tree_divide

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define MAXN (100000+10)
#define MAXM (200000+10)
long long mul(long long a,long long b){return (a*b)%F;}
long long add(long long a,long long b){return (a+b)%F;}
long long sub(long long a,long long b){return
 ↪  (a-b+(a-b)/F*F+F)%F;}
int n;
int edge[MAXM],next[MAXM]={0},pre[MAXN]={0},size=0;
void addedge(int u,int v)
{
  edge[++size]=v;
  next[size]=pre[u];
  pre[u]=size;
}
void addedge2(int u,int v){addedge(u,v),addedge(v,u);}
int opt[MAXN],siz[MAXN],val[MAXN]={0},all[MAXN],tot=0;
void dfs(int x,int fa)
{
  siz[x]=1;opt[x]=0;all[++tot]=x;
  Forp(x)
  {
```

```
    int &v=edge[p];
    if (v^fa&&!val[v])
    {
      dfs(v,x);siz[x]+=siz[v];
      opt[x]=max(opt[x],siz[v]);
    }
  }
}
void solve(int root,int l)
{
  tot=0,dfs(root,0);
  int minopt=INF,minoptx=0;
  For(i,tot)
  {
    int u=all[i];
    opt[u]=max(opt[u],tot-siz[u]);
    if (minopt>opt[u]) minopt=opt[u],minoptx=u;
  }
  val[root=minoptx]=l;
  Forp(root)
  {
    int &v=edge[p];
    if (!val[v]) solve(v,l+1);
  }

}
int main()
{
//  freopen("Commander.in","r",stdin);
  cin>>n;
  For(i,n-1)
  {
    int u,v;
    scanf("%d%d",&u,&v);
    addedge2(u,v);
  }
  solve(1,1);
  For(i,n) if (val[i]>26) {cout<<"Impossible!"<<endl;return 0;}
  For(i,n-1) cout<<(char)(val[i]+'A'-1)<<' ';
  cout<<(char)(val[n]+'A'-1)<<endl;
  return 0;
}
```

”

1.68 tree_divide_hash

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
#include<iomanip>
#include<vector>
#include<string>
#include<queue>
#include<stack>
#include<map>
#include<sstream>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %I64d\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
```

```
#pragma comment(linker, "/STACK:102400000,102400000")
#define F (1000003)
#define MAXN (100000+10)
#define MAXM (200000+10)
long long mul(long long a,long long b){return (a*b)%F;}
long long add(long long a,long long b){return (a+b)%F;}
long long sub(long long a,long long b){return
↪  (a-b+(a-b)/F*F+F)%F;}
typedef long long ll;
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
int n;
int edge[MAXM],Next[MAXM]={0},pre[MAXN]={0},size=0;
void addedge(int u,int v)
{
  edge[++size]=v;
  Next[size]=pre[u];
  pre[u]=size;
}
void addedge2(int u,int v){addedge(u,v),addedge(v,u);}
int v[MAXN],K;
int opt[MAXN],siz[MAXN],val[MAXN]={0},all[MAXN],tot=0;
void dfs(int x,int fa)
{
  siz[x]=1;opt[x]=0;all[++tot]=x;
  Forp(x)
  {
    int &v=edge[p];
    if (v^fa&&!val[v])
    {
      dfs(v,x);siz[x]+=siz[v];
      opt[x]=max(opt[x],siz[v]);
    }
  }
}
int h[F]={0},b[F]={0};
int cas=0;
ll pow2(ll a,int b,ll p)  //a^b mod p
{
    if (b==0) return 1%p;
    if (b==1) return a%p;
```

```
    ll c=pow2(a,b/2,p)%p;
    c=c*c%p;
    if (b&1) c=c*a%p;
    return c%p;
}
ll Inv(ll a,ll p) { //gcd(a,p)=1
  return pow2(a,p-2,p);
}
ll inv[F];
void prework() {
  Rep(i,F) inv[i]=Inv(i,F);
}
void dfs_calc(int x,int fa,ll t,int cas)
{
  t=mul(t,v[x]);
  if (b[t]!=cas||h[t]>x) h[t]=x,b[t]=cas;
  Forp(x)
  {
    int &v=edge[p];
    if (v^fa&&!val[v])
    {
      dfs_calc(v,x,t,cas);

    }
  }
}
pi ans;
void upd(pi &v,pi t) {
  if (t.se==t.fi) return;
  if (t.se<t.fi) swap(t.se,t.fi);
  v=min(v,t);
}
void dfs_calc2(int x,int fa,ll t,int cas)
{
  t=mul(t,inv[v[x]]);
  if (b[t]==cas) {
    upd(ans,mp(x,h[t]));
  }
  Forp(x)
  {
    int &v=edge[p];
    if (v^fa&&!val[v])
    {
      dfs_calc2(v,x,t,cas);

    }
```

```
  }
}
void solve(int root,int l)
{
  tot=0,dfs(root,0);
  int minopt=INF,minoptx=0;
  For(i,tot)
  {
    int u=all[i];
    opt[u]=max(opt[u],tot-siz[u]);
    if (minopt>opt[u]) minopt=opt[u],minoptx=u;
  }
  val[root=minoptx]=l;

  ++cas;
//  b[v[root]]=cas; h[v[root]]=root;
  Forp(root) {
    int &V=edge[p];
    if (!val[V]) {
      dfs_calc2(V,root,K,cas);
      dfs_calc(V,root,v[root],cas);
    }
  }
  if (b[K]==cas) upd(ans,mp(root,h[K]));

  Forp(root)
  {
    int &v=edge[p];
    if (!val[v]) solve(v,l+1);
  }
}
int main()
{
//  freopen("D.in","r",stdin);
  prework();
  while(cin>>n>>K) {
    MEM(edge) MEM(Next) MEM(pre) size=0;
    MEM(opt) MEM(siz) MEM(val)
    For(i,n) v[i]=read();
    For(i,n-1)
    {
      int u,v;
      scanf("%d%d",&u,&v);
      addedge2(u,v);
    }
    ans=mp(INF,INF);
```

```
182       solve(1,1);
183       if (ans==mp(INF,INF)) puts("No solution");
184       else {
185         printf("%d %d\n",ans.fi,ans.se);
186       }
187
188     }
189     return 0;
190   }
```

”

1.69 tree_isomorphism

```
#include<cstdio>
#include<iostream>
#include<string>
#include<vector>
#include<algorithm>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case %d: %lld\n",kcase,ans);
typedef long long ll;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
#define MAXN (3000+10)
string s1="same",s2="different";
struct Str{
```

```
45    vector<int> e[MAXN];
46    string s,ans;
47    int len,tot;
48    void mem() {
49      s='0' + s + '1';
50      len=s.size();
51      tot=len/2;
52      For(i,tot) e[i].clear();
53      tot=0;
54    }
55    int build(int &x) { // suppose s[x]=0
56      int now= ++tot;
57      ++x;
58      while (s[x]!='1') {
59        e[now].pb(build(x));
60      }
61      x++;
62      return now;
63    }
64    string get_str(int x) {
65      vector<string> tmp;
66      int m=SI(e[x]);
67      Rep(i,m) tmp.pb(get_str(e[x][i]));
68      sort(tmp.begin(),tmp.end());
69      string ans;
70      Rep(i,m) ans+=(tmp[i]);
71      return '0'+ans +'1';
72    }
73  }t[2];
74  int main()
75  {
76  //  freopen(".in","r",stdin);
77    int T=read();
78    For(kcase,T) {
79      cin>>t[0].s >> t[1].s;
80      t[0].mem(); t[1].mem();
81      int p;
82      t[0].build(p=0);
83      t[1].build(p=0);
84      if (t[0].get_str(1) == t[1].get_str(1) ) cout<<s1;
85      else cout<<s2;
86      cout<<endl;
87
88
89    }
90
```

```
91
92      return 0;
93  }
```

”

1.70 Trie

```cpp
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (x<<1)
#define Rson ((x<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (20071027)
long long mul(long long a,long long b){return (a*b)%F;}
long long add(long long a,long long b){return (a+b)%F;}
long long sub(long long a,long long b){return
 ↪  (a-b+(a-b)/F*F+F)%F;}
typedef long long ll;
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
#define MAXNode (100000*5+10)
#define Sigma_size (26)
class Trie
{
public:
  int ch[MAXNode][Sigma_size];
  int v[MAXNode],siz;
  Trie(int _siz=0):siz(_siz){MEM(ch) MEM(v)}
  void mem(int _siz=0){siz=_siz; MEM(ch) MEM(v)  }
  int idx(char c){return c-'a';}
  void insert(char *s,int val=0)
  {
    int u=0,n=strlen(s);
    Rep(i,n)
```

```
44      {
45        int c=idx(s[i]);
46        if (!ch[u][c])
47        {
48          ++siz;
49          MEM(ch[siz]);
50          ch[u][c]=siz;
51        }
52        u=ch[u][c];
53      }
54      v[u]=val;
55    }
56    void find(char *s)
57    {
58      int u=0,n=strlen(s);
59      Rep(i,n)
60      {
61        int c=idx(s[i]);
62        if (!ch[u][c])
63        {
64          return;
65        }
66        u=ch[u][c];
67      }
68    }
69
70  }T;
71  int main()
72  {
73  //  freopen(".in","r",stdin);
74  //  freopen(".out","w",stdout);
75
76
77    return 0;
78  }
```

”

1.71 YenAlgorithm

```cpp
#include <bits/stdc++.h>
using namespace std;

typedef vector<int> Path;

struct Cell{
  int len,pos;
  Path path,revPath;
  vector< vector<int> > forbiddenList;
  Cell(){}
  Cell(int _len,int _pos,Path _path) :
   ↪  len(_len),pos(_pos),path(_path){
    revPath=path;
    reverse(path.begin(),path.end());
  }
    bool operator <(const Cell &t) const{return len>t.len ||
     ↪  len==t.len && revPath>t.revPath;}
};
#define RESET()\
    memset(pre,0,sizeof(pre)),\
    memset(d,0x3f,sizeof(d)),\
    memset(flag,0,sizeof(flag)),\
    d[S]=0,flag[S]=1

const int INF =0x3f3f3f3f;

int n,m,K,s,t;
int g[55][55];
int pre[55],d[55];
int flag[55];
int filter[55][55];

priority_queue<Cell> q;

Cell dijkstra(int S,int T) {
  int now=S;
  while(1) {
    flag[now]=1;
    for(int i=1;i<=n;i++) if (!filter[now][i] &&g[now][i]<INF) {
      if (g[now][i]+d[now]<d[i]) d[i]=d[now]+g[now][i],pre[i]=now;
      else if (g[now][i]+d[now]==d[i]&& now<pre[i]) pre[i]=now;
    }
    now=0;
    for(int i=1;i<=n;i++) if (!flag[i] && d[i]<d[now]) now=i;
```

```
43      if (!now) break;
44    }
45    if (d[T]==INF) return Cell(0,0,Path());
46    int fork_p,cnt=0;Path tmp;
47    for(int p=T;p;p=pre[p],cnt++)
      ↪  tmp.push_back(p),pre[p]==S&&(fork_p =cnt);
48    return Cell(d[T],tmp.size()-1-fork_p,tmp);
49  }
50  void modify(const Cell &pre,Cell &now) {
51    for(int i=0;i<now.path.size();i++)
      ↪  now.forbiddenList.push_back(vector<int>() );
52    for(int i=0;i<now.path.size()-1;i++)
      ↪  now.forbiddenList[i].push_back(now.path[i+1]) ;
53    if (pre.path.empty()) return;
54    now.forbiddenList[now.pos-1]=pre.forbiddenList[now.pos-1];
55    now.forbiddenList[now.pos-1].push_back(now.path[now.pos]);
56  }
57  void printPath(Path &path) {
58    if (!path.size()) puts("No");
59    else {
60      printf("%d",path[0]);
61      for(int i=1;i<path.size();i++) printf("-%d",path[i]);
62      puts("");
63    }
64  }
65  Path yenAlgorithm(int S,int T,int K) {
66    RESET(); Cell now=dijkstra(S,T);
67    if (now.path.empty()) return Path();
68    modify(Cell(),now);
69    for(int i=1;i<K;i++) {
70      Path nowP=now.path;
71      int pos=now.pos;
72      for(int j=pos-1;j<nowP.size()-1;j++) {
73        RESET();
74        for(int k=1;k<=j;k++) d[nowP[k]]=d[pre[nowP[k]]=nowP[k-
          ↪  1]]+g[nowP[k-1]][nowP[k]],flag[nowP[k]]=1;
75        memset(filter,0,sizeof(filter));
76        for(int k=0;k<now.forbiddenList[j].size();k++)
          ↪  filter[nowP[j]][now.forbiddenList[j][k]]=1;
77        Cell newOne=dijkstra(nowP[j],T);
78        if (newOne.path.empty()) continue;
79        modify(now,newOne);
80        q.push(newOne);
81       }
82      if(q.empty()) return Path();
83      now=q.top();q.pop();
```

```
    }
    return now.revPath;
}
int main(){
//  freopen("bzoj1073.in","r",stdin);
    cin>>n>>m>>K>>s>>t;
    memset(g,0x3f,sizeof(g));
    while(m--) {
      int u,v,w;
      cin>>u>>v>>w;
      g[v][u]=w;
    }
    Path path = yenAlgorithm(t,s,K);
    printPath(path);
    return 0;
}
```

”

2 GoodProblem

2.1 BigInteger

```cpp
#include <algorithm> // max
#include <cassert>   // assert
#include <cstdio>    // printf,sprintf
#include <cstring>   // strlen
#include <iostream>  // cin,cout
#include <string>    // string 类
#include <vector>    // vector 类
using namespace std;

struct BigInteger {
    typedef unsigned long long LL;

    static const int BASE = 100000000;
    static const int WIDTH = 8;
    vector<int> s;

    BigInteger& clean(){while(!s.back()&&s.size()>1)s.pop_back();
    ↪  return *this;}
    BigInteger(LL num = 0) {*this = num;}
    BigInteger(string s) {*this = s;}
    BigInteger& operator = (long long num) {
        s.clear();
        do {
            s.push_back(num % BASE);
            num /= BASE;
        } while (num > 0);
        return *this;
    }
    BigInteger& operator = (const string& str) {
        s.clear();
        int x, len = (str.length() - 1) / WIDTH + 1;
        for (int i = 0; i < len; i++) {
            int end = str.length() - i*WIDTH;
            int start = max(0, end - WIDTH);
            sscanf(str.substr(start,end-start).c_str(), "%d", &x);
            s.push_back(x);
        }
        return (*this).clean();
    }

    BigInteger operator + (const BigInteger& b) const {
```

```
41          BigInteger c; c.s.clear();
42          for (int i = 0, g = 0; ; i++) {
43              if (g == 0 && i >= s.size() && i >= b.s.size()) break;
44              int x = g;
45              if (i < s.size()) x += s[i];
46              if (i < b.s.size()) x += b.s[i];
47              c.s.push_back(x % BASE);
48              g = x / BASE;
49          }
50          return c;
51      }
52      BigInteger operator - (const BigInteger& b) const {
53          assert(b <= *this); // 减数不能大于被减数
54          BigInteger c; c.s.clear();
55          for (int i = 0, g = 0; ; i++) {
56              if (g == 0 && i >= s.size() && i >= b.s.size()) break;
57              int x = s[i] + g;
58              if (i < b.s.size()) x -= b.s[i];
59              if (x < 0) {g = -1; x += BASE;} else g = 0;
60              c.s.push_back(x);
61          }
62          return c.clean();
63      }
64      BigInteger operator * (const BigInteger& b) const {
65          int i, j; LL g;
66          vector<LL> v(s.size()+b.s.size(), 0);
67          BigInteger c; c.s.clear();
68          for(i=0;i<s.size();i++) for(j=0;j<b.s.size();j++)
            ↪ v[i+j]+=LL(s[i])*b.s[j];
69          for (i = 0, g = 0; ; i++) {
70              if (g ==0 && i >= v.size()) break;
71              LL x = v[i] + g;
72              c.s.push_back(x % BASE);
73              g = x / BASE;
74          }
75          return c.clean();
76      }
77      BigInteger operator / (const BigInteger& b) const {
78          assert(b > 0);  // 除数必须大于 0
79          BigInteger c = *this;       // 商: 主要是让 c.s 和
            ↪ (*this).s 的 vector 一样大
80          BigInteger m;               // 余数: 初始化为 0
81          for (int i = s.size()-1; i >= 0; i--) {
82              m = m*BASE + s[i];
83              c.s[i] = bsearch(b, m);
```

```
84          m -= b*c.s[i];
85          }
86          return c.clean();
87      }
88      BigInteger operator % (const BigInteger& b) const { //方法与除
        ↪ 法相同
89          BigInteger c = *this;
90          BigInteger m;
91          for (int i = s.size()-1; i >= 0; i--) {
92              m = m*BASE + s[i];
93              c.s[i] = bsearch(b, m);
94              m -= b*c.s[i];
95          }
96          return m;
97      }
98      // 二分法找出满足 bx<=m 的最大的 x
99      int bsearch(const BigInteger& b, const BigInteger& m) const{
100         int L = 0, R = BASE-1, x;
101         while (1) {
102             x = (L+R)>>1;
103             if (b*x<=m) {if (b*(x+1)>m) return x; else L = x;}
104             else R = x;
105         }
106     }
107     BigInteger& operator += (const BigInteger& b) {*this = *this +
        ↪ b; return *this;}
108     BigInteger& operator -= (const BigInteger& b) {*this = *this -
        ↪ b; return *this;}
109     BigInteger& operator *= (const BigInteger& b) {*this = *this *
        ↪ b; return *this;}
110     BigInteger& operator /= (const BigInteger& b) {*this = *this /
        ↪ b; return *this;}
111     BigInteger& operator %= (const BigInteger& b) {*this = *this %
        ↪ b; return *this;}
112
113     bool operator < (const BigInteger& b) const {
114         if (s.size() != b.s.size()) return s.size() < b.s.size();
115         for (int i = s.size()-1; i >= 0; i--)
116             if (s[i] != b.s[i]) return s[i] < b.s[i];
117         return false;
118     }
119     bool operator >(const BigInteger& b) const{return b < *this;}
120     bool operator<=(const BigInteger& b) const{return !(b <
        ↪ *this);}
121     bool operator>=(const BigInteger& b) const{return !(*this <
        ↪ b);}
```

```
122     bool operator!=(const BigInteger& b) const{return b < *this ||
        ↪  *this < b;}
123     bool operator==(const BigInteger& b) const{return !(b < *this)
        ↪  && !(b > *this);}
124 };
125
126 ostream& operator << (ostream& out, const BigInteger& x) {
127     out << x.s.back();
128     for (int i = x.s.size()-2; i >= 0; i--) {
129         char buf[20];
130         sprintf(buf, "%08d", x.s[i]);
131         for (int j = 0; j < strlen(buf); j++) out << buf[j];
132     }
133     return out;
134 }
135
136 istream& operator >> (istream& in, BigInteger& x) {
137     string s;
138     if (!(in >> s)) return in;
139     x = s;
140     return in;
141 }
142 int main() {
143   BigInteger a,b;
144   a=1; b=1;
145   cout<<(a+b)<<endl;
146   return 0;
147 }
```

”

2.2 bzoj1061_Linear_Programming

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,0x3f,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEMx(a,b) memset(a,b,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
#define ALL(x) (x).begin(),(x).end()
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
```

```
}
int m,n;
#define EPS (1E-7)
#define MAXM (10000+10)
#define MAXN (1000+10)
namespace Linear_Programming{
    double A[MAXM][MAXN],b[MAXM],c[MAXM],v;
    void Pivot(int l,int e)
    {
        int i,j;
        b[l]/=A[l][e];
        for(i=1;i<=n;i++)
            if(i!=e)
                A[l][i]/=A[l][e];
        A[l][e]=1/A[l][e];

        for(i=1;i<=m;i++)
            if(i!=l&&fabs(A[i][e])>EPS)
            {
                b[i]-=A[i][e]*b[l];
                for(j=1;j<=n;j++)
                    if(j!=e)
                        A[i][j]-=A[i][e]*A[l][j];
                A[i][e]=-A[i][e]*A[l][e];
            }

        v+=c[e]*b[l];
        for(i=1;i<=n;i++)
            if(i!=e)
                c[i]-=c[e]*A[l][i];
        c[e]=-c[e]*A[l][e];
    }
    double Simplex()
    {
        int i,l,e;
        while(1)
        {
            for(i=1;i<=n;i++)
                if(c[i]>EPS)
                    break;
            if((e=i)==n+1)
                return v;
            double temp=INF;
            for(i=1;i<=m;i++)
                if( A[i][e]>EPS && b[i]/A[i][e]<temp )
                    temp=b[i]/A[i][e],l=i;
```

```
91              if(temp==INF) return INF;
92              Pivot(l,e);
93          }
94    }
95  }
96  int main()
97  {
98  //  freopen("bzoj1061.in","r",stdin);
99  //  freopen(".out","w",stdout);
100
101   using namespace Linear_Programming;
102   n=read(),m=read();
103   For(i,n) { //n= 方程数 m= 变元数
104     scanf("%lf",&c[i]);
105   }
106   For(i,m) {
107     int x=read(),y=read(),z=read();
108     for(int j=x;j<=y;j++) A[i][j]=1;
109     b[i]=z;
110   }
111   double ans=Simplex();
112     printf("%d\n",int(ans+0.5));
113   return 0;
114 }
```

”

2.3 bzoj1103_Artificial_Stack_dfs

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %I64d\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
```

```
}
int n,m;
struct BIT{
  #define MAXN ( 2*250000+10)
  ll f[MAXN];
  void add(int x,ll v) {
    for(int i=x;i<=2*n;i+=i&(-i))
         f[i]+=v;
  }
  ll qur(int x) {
    ll v=0;
    for(int i=x;i;i-=i&(-i))
         v+=f[i];
    return v;
  }
}T;
int tim=0;
ll a[MAXN];
vi edges[MAXN];
int st[MAXN]={0},top=0;
int l[MAXN]={0},r[MAXN],fa[MAXN]={0};
void dfs() {
  st[++top]=1;
  while(top) {
    int now=st[top];
    top--;
    if (!l[now]) {
      l[now]=++tim;
      st[++top]=now;
      Rep(i,SI(edges[now])) {
        int v=edges[now][i];
        if (v^fa[now]) {
          st[++top]=v;
          fa[v]=now;
        }
      }
    } else r[now]=++tim;
  }
}

int main()
{
//  freopen("bzoj1103.in","r",stdin);
//  freopen(".out","w",stdout);
  n=read();
  For(i,n-1) {
```

```
    int u=read(),v=read();
    edges[u].pb(v);
    edges[v].pb(u);
  }
  dfs();
  MEM(T.f)
  For(i,n) {
    T.add(l[i],1);T.add(r[i],-1);
  }
  m=read();
  For(i,n+m-1) {
    char op[2];
    scanf("%s",op);
    if(op[0]=='W') {
      printf("%lld\n",T.qur(l[read()])-1);
    } else {
      int u=read(),v=read();
      if (l[u]>l[v]) swap(u,v);
      T.add(l[v],-1);T.add(r[v],1);
    }
  }
  return 0;
}
```

”

2.4 bzoj1876_std

```cpp
#include <cstdio>
#include <cstring>
#include <algorithm>
#define rep(i,l,r) for (int i=l;i<=r;++i)
const int Mx=1252,MOD=100000000;
struct BIGN{
  int a[Mx+10];
  BIGN(){memset(a,0,sizeof a);}
  int &operator [](int i){return a[i];}
  void operator /=(int x){
    for (int i=Mx;i>=1;--i)
      a[i-1]+=a[i]%x*MOD,a[i]/=x;
    }
  void operator -=(BIGN &b){
    for (int i=1;i<Mx;++i)
      a[i]=a[i]-b[i]+(a[i-1]+MOD)/MOD -1,a[i-1]=(a[i-1]+MOD)%MOD;
    }
  void operator *=(int x){
    for (int i=1;i<Mx;++i)
      a[i]=a[i]*x+a[i-1]/MOD,a[i-1]%=MOD;
    }
  bool operator <(BIGN &b){
    for (int i=Mx;i>=1;--i)
      if (a[i]!=b[i]) return a[i]<b[i];
    return false;
    }
  bool iszero(){
    for (int i=1;i<Mx;++i) if (a[i]!=0) return false;
    return true;
    }
  void read(){
    char tp[10005]={'0','0','0','0','0','0','0','0'};
    scanf("%s",tp+8);
    int len=strlen(tp+1),p=1;
    while (len-8*p+1>0)
      sscanf(tp+len-8*p+++1,"%8d",&a[p]);
    }
  void print(){
    int p=Mx;
    while (!a[p]&&p>0) p--;
    printf("%d",a[p--]);
    while (p>0) printf("%08d",a[p--]);
    printf("\n");
    }
```

```
};
BIGN gcd(BIGN x,BIGN y){
  int g=0;bool x1,y1;
  while (!x.iszero() && !y.iszero()){
    x1=!(x[1]&1),y1=!(y[1]&1);
    if (x1 && y1){g++;x/=2,y/=2;}else
    if (x1 || y1){if (x1) x/=2;else y/=2;}else
    if (y<x) x-=y;else y-=x;
    }
  if (x<y) x=y;
  while (g--) x*=2;
  return x;
}
BIGN a,b;
int main(){
  freopen("bzoj1876.in","r",stdin);
  a.read();
  b.read();
  gcd(a,b).print();
}
```

”

2.5 bzoj2243-染色

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,0x3f,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEMx(a,b) memset(a,b,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
```

```
45    return x*f;
46  }
47  #define MAXN (100001+10)
48  ll cover[MAXN<<2],ls[MAXN<<2],rs[MAXN<<2],sum[MAXN<<2];
49  void pushUp(int o,int m) {
50    ls[o]=ls[Lson];
51    rs[o]=rs[Rson];
52    sum[o]=sum[Lson]+sum[Rson]-(rs[Lson]==ls[Rson]);
53  }
54  void pushDown(int o,ll m) {
55    if(m<=1) return;
56    if (cover[o]!=-1) {
57      cover[Lson]=cover[Rson]=cover[o];
58      ls[Lson]=rs[Lson]=ls[Rson]=rs[Rson]=cover[o];
59      sum[Lson]=sum[Rson]=1;
60      cover[o]=-1;
61    }
62  }
63  void build(int l,int r,int o) {
64    cover[o]=-1;
65    sum[o]=ls[o]=rs[o]=1;
66    if (l==r) {
67      return;
68    }
69    int m=(l+r)>>1;
70    build(l,m,Lson);
71    build(m+1,r,Rson);
72    pushUp(o,r-l+1);
73  }
74  void update(int l,int r,int o,int L,int R,int c) {
75    if (L<=l&&r<=R) {
76      if (l<r) cover[o]=c;
77      ls[o]=rs[o]=c;
78      sum[o]=1;
79      return;
80    }
81    pushDown(o,r-l+1);
82    int m=(l+r)>>1;
83    if (L<=m) update(l,m,Lson,L,R,c);
84    if (m<R) update(m+1,r,Rson,L,R,c);
85    pushUp(o,r-l+1);
86  }
87  int query(int l,int r,int o,int L,int R) {
88    if (L<=l&&r<=R) return sum[o];
89    pushDown(o,r-l+1);
90    int m=(l+r)>>1;
```

```
  int ret=0;
  if (L<=m) ret+=query(l,m,Lson,L,R);
  if (m<R) ret+=query(m+1,r,Rson,L,R);
  if (L<=m&&m<R) ret-=rs[Lson]==ls[Rson];
  return ret;
}
int queryc(int l,int r,int o,int x) {
  if (l==r) return ls[o];
  if (cover[o]!=-1) return cover[o];
  int m=(l+r)>>1;
  if (x<=m) return queryc(l,m,Lson,x);
  return queryc(m+1,r,Rson,x);
}
int n,m;
struct Tree{
  #define MAXM (100001*2)
  void mem(){MEM(Pre) siz=1;}
  int edge[MAXM],Next[MAXM],Pre[MAXN],siz;
  void addedge(int u,int v)
  {
    edge[++siz]=v;
    Next[siz]=Pre[u];
    Pre[u]=siz;
  }
  void addedge2(int u,int v){addedge(u,v);addedge(v,u);}
  bool vis[MAXN];
  int cnt,id[MAXN];
  int son[MAXN],dep[MAXN],sz[MAXN],top[MAXN],pre[MAXN],q[MAXN];
  void build()
  {
    MEM(vis) cnt=0; MEM(id)
    MEM(son) MEM(dep) MEM(sz) MEM(top) MEM(pre) MEM(q)
    int r=1;
    vis[dep[1]=q[1]=1]=1;
    For(i,r)
    {
      int u=q[i];
      Forp(u)
      {
        int v=edge[p];
        if (vis[v]) continue; else vis[v]=1;
        dep[ q[++r]=v ]=dep[u]+1;
        pre[v]=u;
      }
    }
    ForD(i,r) {
```

```
137         sz[pre[q[i]]] += ++sz[q[i]];
138         if (sz[son[pre[q[i]]]]<sz[q[i]] ) son[pre[q[i]]] = q[i];
139       }
140       For(i,r) {
141         if (!top[q[i]])
142           for(int x=q[i];x;x=son[x]) {
143             top[x]=q[i];
144             id[x]=++cnt;
145           }
146       }
147 
148     }
149     int lca(int a,int b)
150     {
151       while(1) {
152         if (top[a]==top[b]) return dep[a]<=dep[b] ? a:b;
153         if (dep[top[a]]<dep[top[b]]) swap(a,b);
154         a=pre[top[a]];
155       }
156     }
157 
158     ll Ask(int a,int b)
159     {
160       ll ans=0;
161       while (top[a]^top[b]) {
162         ans+=query(1,n,1,id[top[a]],id[a]);
163         ans-
            ↪  =queryc(1,n,1,id[top[a]])==queryc(1,n,1,id[pre[top[a]]]);
164         a=pre[top[a]];
165       }
166       ans+=query(1,n,1,id[b],id[a]);
167       return ans;
168     }
169     void set(int a,int b,int c)
170     {
171       while (top[a]^top[b]) {
172         update(1,n,1,id[top[a]],id[a],c);
173         a=pre[top[a]];
174       }
175       update(1,n,1,id[b],id[a],c);;
176     }
177 
178   }S;
179   int a[MAXN];
180   int main()
181   {
```

```
182 //  freopen("bzoj2243.in","r",stdin);
183 //  freopen("bzoj2243.out","w",stdout);
184   n=read(),m=read();
185   build(1,n,1);
186   S.mem();
187   For(i,n) a[i]=read();
188   For(i,n-1) S.addedge2(read(),read());
189   S.build();
190   For(i,n) S.set(i,i,a[i]);
191 
192   while(m--) {
193     char op[2]; int a,b;
194     scanf("%s%d%d",op,&a,&b);
195     int t=S.lca(a,b);
196     if (op[0]=='Q') {
197        printf("%lld\n",S.Ask(a,t)+S.Ask(b,t)-1);
198     } else {
199       int c;
200       scanf("%d",&c);
201       S.set(a,t,c);
202       S.set(b,t,c);
203     }
204   }
205   return 0;
206 }
```

”

2.6 Bzoj2595_Steiner_Tree

```cpp
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,0x3f,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEMx(a,b) memset(a,b,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
#define ALL(x) (x).begin(),(x).end()
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
```

```
45    return x*f;
46  }
47  #define MAXN (1100)
48  int n,m,K;
49  vi edges[MAXN],weight[MAXN];
50  int bo[MAXN]={},dp[MAXN][1<<10],w[MAXN]={},f[MAXN];
51  queue<int> q;
52  int pd[MAXN]={};
53  void spfa(int now) {
54    For(i,n) q.push(i),pd[i]=1;
55    while(!q.empty()) {
56      int x=q.front(); q.pop();
57      Rep(i,SI(edges[x])) {
58        int v=edges[x][i];
59        if (dp[x][now]+weight[x][i]<dp[v][now]) {
60          dp[v][now]=dp[x][now]+weight[x][i];
61          if (!pd[v]) pd[v]=1,q.push(v);
62        }
63      }
64      pd[x]=0;
65    }
66  }
67  int main()
68  {
69  //  freopen("bzoj4006.in","r",stdin);
70  //  freopen(".out","w",stdout);
71    cin>>n>>m>>K;
72    For(i,m) {
73      int u=read(),v=read(),w=read();
74      edges[u].pb(v);  edges[v].pb(u);
75      weight[u].pb(w); weight[v].pb(w);
76    }
77    For(i,K) {
78      int c=read(),id=read();
79      bo[id]=i; w[c]+=(1<<i-1);
80    }
81    int S=1<<K;
82    MEMI(dp)
83    For(now,S-1) {
84      For(i,n) {
85        if (bo[i] && (1<<bo[i]-1) == now ) dp[i][now]=0;
86        for(int j=(now-1)&now;j;j=(j-1)&now)
          ↪  dp[i][now]=min(dp[i][now],dp[i][j]+dp[i][now-j]);
87      }
88      spfa(now);
89    }
```

```
  MEMI(f)
  For(i,S-1) {
    int k=0;
    For(j,K) if ((i>>j-1)&1) k|=w[j];
    For(j,n) f[i]=min(f[i],dp[j][k]);
    for(int j=(i-1)&i;j;j=(j-1)&i) f[i]=min(f[i],f[j]+f[i-j]);
  }
  cout<<f[S-1];
  return 0;
}
```

”

2.7 bzoj3329_Xorequ_nth_Fibonacci

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,0x3f,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEMx(a,b) memset(a,b,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
#define ALL(x) (x).begin(),(x).end()
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
```

```
  return x*f;
}
ll DP[100][2][2];
ll DIG[100];
ll dfs(int pos,int pre,int status,int limit) // pos from high to
 ↪  low
{
  if(pos < 1) {
    return !status;
  }
    if(!limit && DP[pos][pre][status] != -1) {
    return DP[pos][pre][status];
    }

  int end = limit ? DIG[pos] : 1;
  ll ret = 0;

  for(int i = 0;i <= end;i ++)
    ret += dfs(pos - 1,i,status || (pre == 1 && i == 1),limit &&
     ↪  (i == end));

  if(!limit)
    DP[pos][pre][status] = ret;
  return  ret;
}
#define MAXN (3)
struct M
{
    int n,m;
    ll a[MAXN][MAXN];
    M(int _n=0){n=m=_n;MEM(a);}
    M(int _n,int _m){n=_n,m=_m;MEM(a);}
  void make_I(int _n)
    {
      n=m=_n; MEM(a)
        For(i,n) a[i][i]=1;
    }
}A;
M operator*(M a,M b)
{
    M c(a.n,b.m);
    For(k,a.m)
      For(i,a.n)
            For(j,b.m)
                c.a[i][j]=(c.a[i][j]+a.a[i][k]*b.a[k][j])%F;
  return c;
```

```
}
M pow2(M a,ll b)
{
    M c;c.make_I(a.n);
    static bool a2[1000000];
    int n=0;while (b) a2[++n]=b&1,b>>=1;
    For(i,n)
    {
        if (a2[i]) c=c*a;
        a=a*a;
    }
    return c;
}
ll work2(ll n) {
  M fib(2,2),ans(1,2);
  fib.a[1][1]=fib.a[1][2]=fib.a[2][1]=1;
  ans.a[1][1]=ans.a[1][2]=1;
  return (ans*pow2(fib,n)).a[1][1];

}
int main()
{
//  freopen("bzoj3329.in","r",stdin);
//  freopen(".out","w",stdout);
  int T=read();
  while(T--) {
    MEMx(DP,-1)
    ll x;cin>>x;
    ll ans2=work2(x);
    int len=0;
    while(x) {
      DIG[++len]=x%2;
      x/=2;
    }
    ll ans1=dfs(len,-1,0,1)-1;
    cout<<ans1<<endl<<ans2<<endl;
  }


  return 0;
}
```

”

2.8 bzoj3680_hillClimbing

```
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,0x3f,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEMx(a,b) memset(a,b,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
#define ALL(x) (x).begin(),(x).end()
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
```

```
  return x*f;
}
#define MAXN (10010)
int n;
double x[MAXN],y[MAXN],g[MAXN];
double sqr(double x){return x*x;}
double dis(double a,double b,int i) {
  return sqrt(sqr(a-x[i])+sqr(b-y[i]));
}
void hillclimbing() {
  double t=1000,X,Y,ansx=0,ansy=0;
  For(i,n) {
    ansx+=x[i]*g[i]; ansy+=y[i]*g[i];
  }
  ansx/=n,ansy/=n;
  while(t>1e-8) {
    X=Y=0;
    For(i,n) {
      X+=(x[i]-ansx)*g[i]/dis(ansx,ansy,i);
      Y+=(y[i]-ansy)*g[i]/dis(ansx,ansy,i);
    }
    ansx+=X*t;
    ansy+=Y*t;
    if (t>0.5) t*=0.5;
    else t*=0.97;
  }
  printf("%.3lf %.3lf\n",ansx,ansy);
}
int main()
{
//  freopen("bzoj3680.in","r",stdin);
//  freopen(".out","w",stdout);

  n=read();
  For(i,n) {
    scanf("%lf%lf%lf",&x[i],&y[i],&g[i]);
  }
  hillclimbing();

  return 0;
}
```

”

2.9 Frightful_Formula_clrs97_fft_mod_1e9+7

```cpp
#include<cstdio>
#include<cmath>
#include<iostream>
#include<algorithm>
using namespace std;
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (1000003)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %I64d\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
const int N=524300,P=1000003,M=1000;
int n,a,b,c,i,j,k,pos[N],ans;
int pa[N],pb[N],fac[N],inv[N],A[N],B[N],C[N];
namespace FFT{
  struct comp{
    ld r,i;comp(ld _r=0,ld _i=0){r=_r;i=_i;}
    comp operator+(const comp x){return comp(r+x.r,i+x.i);}
```

```
45      comp operator-(const comp x){return comp(r-x.r,i-x.i);}
46      comp operator*(const comp x){return
          ↪  comp(r*x.r-i*x.i,r*x.i+i*x.r);}
47      comp conj(){return comp(r,-i);}
48    }A[N],B[N];
49    int a0[N],b0[N],a1[N],b1[N];
50    const ld pi=acos(-1.0);
51    void FFT(comp a[],int n,int t){
52      For(i,n-1) if(i<pos[i]) swap(a[i],a[pos[i]]);
53      for(int d=0;(1<<d)<n;d++) {
54          int m=1<<d,m2=m<<1;
55          ld o=pi*2/m2*t;comp _w(cos(o),sin(o));
56          for(int i=0;i<n;i+=m2){
57          comp w(1,0);
58          for(int j=0;j<m;j++){
59            comp&A=a[i+j+m],&B=a[i+j],t=w*A;
60            A=B-t;B=B+t;w=w*_w;
61          }
62        }
63      }
64      if(t==-1)for(int i=0;i<n;i++)a[i].r/=n;
65    }
66    //c=a*b
67    void mul(int*a,int*b,int*c){
68      int i,j;
69      for(i=0;i<k;i++)A[i]=comp(a[i],b[i]);
70      FFT(A,k,1);
71      for(i=0;i<k;i++){
72        j=(k-i)&(k-1);
73        B[i]=(A[i]*A[i]-(A[j]*A[j]).conj())*comp(0,-0.25);
74      }
75      FFT(B,k,-1);
76      for(i=0;i<k;i++)c[i]=((long long)(B[i].r+0.5))%P;
77    }
78    //c = a*b mod P
79    void mulmod(int*a,int*b,int*c){
80      int i;
81      for(i=0;i<k;i++)a0[i]=a[i]/M,b0[i]=b[i]/M;
82      for(mul(a0,b0,a0),i=0;i<k;i++){
83        c[i]=1LL*a0[i]*M*M%P;
84        a1[i]=a[i]%M,b1[i]=b[i]%M;
85      }
86      for(mul(a1,b1,a1),i=0;i<k;i++){
87        c[i]=(a1[i]+c[i])%P,a0[i]=(a0[i]+a1[i])%P;
88        a1[i]=a[i]/M+a[i]%M,b1[i]=b[i]/M+b[i]%M;
89      }
```

```
    for(mul(a1,b1,a1),i=0;i<k;i++)c[i]=(1LL*M*(a1[i]-a0[i]+P)+c[i])%P;
  }
}
int main(){
//  freopen("F.in","r",stdin);
  while(cin>>n>>a>>b>>c) {
    ans=0;
    for(pa[0]=i=1;i<=n;i++)pa[i]=1LL*pa[i-1]*a%P;
    for(pb[0]=i=1;i<=n;i++)pb[i]=1LL*pb[i-1]*b%P;
    for(fac[0]=i=1;i<=n+n;i++)fac[i]=1LL*fac[i-1]*i%P;
    for(inv[0]=inv[1]=1,i=2;i<=n;i++)inv[i]=1LL*(P-inv[P%i])*(P/i)%P;
    for(i=1;i<=n;i++)inv[i]=1LL*inv[i]*inv[i-1]%P;
    for(i=1;i<=n;i++){
      cin>>j;
      if(i>1)ans=(1LL*fac[n+n-i-2]*inv[n-i]%P*pa[n-1]%P*pb[n-i]%P*j+ans)%P;
     }
    for(i=1;i<=n;i++){
      cin>>j;
      if(i>1)ans=(1LL*fac[n+n-i-2]*inv[n-i]%P*pa[n-i]%P*pb[n-1]%P*j+ans)%P;
    }
    ans=1LL*ans*inv[n-2]%P;
    for(k=1;k<=n;k<<=1);k<<=1;
    j=__builtin_ctz(k)-1;
    for(i=0;i<k;i++)pos[i]=pos[i>>1]>>1|((i&1)<<j);
    for(i=2;i<=n;i++)A[i]=1LL*pa[n-i]*inv[n-i]%P;
    for(i=2;i<=n;i++)B[i]=1LL*pb[n-i]*inv[n-i]%P;
    FFT::mulmod(A,B,C);
    for(i=4;i<=n+n;i++)ans=(1LL*C[i]*fac[n+n-i]%P*c+ans)%P;
    printf("%d\n",ans);
  }
  return 0;
}
```

”

2.10 hdu5915_Basering_Tree_Plus_Dp

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
#include<functional>
#include<iostream>
#include<cmath>
#include<cctype>
#include<ctime>
#include<iomanip>
#include<bitset>
#include<vector>
#include<string>
#include<queue>
#include<stack>
#include<map>
#include<sstream>
#include<complex>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,0x3f,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEMx(a,b) memset(a,b,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define vpi vector<pi >
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
```

```
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
#define ALL(x) (x).begin(),(x).end()
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
  return x*f;
}
#define MAXN (101000)
int n,p[MAXN];
vi edges[MAXN];
bitset<MAXN> used,inCircle;
int circleLen,circleHead,circletail;
void init_dfs(int x,int fa) {
  p[x]=fa;
  used[x]=1;
  for(auto v:edges[x]) if (v!=fa) {
    if (used[v]) {
      circleHead=x;
      circletail=v;
      continue;
    }
    init_dfs(v,x);
  }
}
int d[MAXN],maxDepth[MAXN],maxDepthP[MAXN];
pi d_p[MAXN];
void dfs(int x,int fa) {
  d[x] = maxDepth[x] = 0;
  d_p[x]=mp(x,x); maxDepthP[x]=x;
  pi tmp_p;
  for(auto v:edges[x]) {
    if (v==fa||inCircle[v]) continue;
    dfs(v,x);
```

```
91      if (d[x] < d[v]) {
92        d[x]=d[v];
93        d_p[x]=d_p[v];
94      } else if (d[x] == d[v] && d_p[x] > d_p[v] ) {
95        d_p[x]=d_p[v];
96      }
97      tmp_p.fi=min(maxDepthP[x],maxDepthP[v]);
98      tmp_p.se=max(maxDepthP[x],maxDepthP[v]);
99      if (d[x]<maxDepth[x]+maxDepth[v]+1) {
100       d[x]=maxDepth[x]+maxDepth[v]+1;
101       d_p[x]=tmp_p;
102     } else if (d[x]==maxDepth[x]+maxDepth[v]+1 && d_p[x] > tmp_p)
        ↪  {
103       d_p[x]=tmp_p;
104     }
105     if (maxDepth[x]<maxDepth[v]+1 || (maxDepth[x] == maxDepth[v]+1
        ↪  && maxDepthP[x]>maxDepthP[v] )) {
106       maxDepth[x]=maxDepth[v]+1;
107       maxDepthP[x] = maxDepthP[v];
108     }
109   }
110 }
111 void MAIN() {
112   n=read();
113   Rep(i,n) edges[i].clear();
114   Rep(i,n) {
115     int u=read(),v=read();
116     u--,v--;
117     edges[u].pb(v);
118     edges[v].pb(u);
119   }
120   used.reset(); inCircle.reset();
121   init_dfs(0,0);
122   vi circle;
123   for(int i=circletail ;;i=p[i]) {
124     inCircle[i]=1;
125     circle.pb(i);
126     if (i==circleHead) break;
127   }
128   int ans=n*2;
129   pi ans_p=mp(0,0);
130   circleLen=SI(circle);
131   for(auto x:circle) {
132     dfs(x,x);
133     if (ans>2*n-circleLen-d[x] || ans==2*n-circleLen-d[x] &&
      ↪  ans_p>d_p[x]) {
```

```
134         ans=2*n-circleLen-d[x];
135         ans_p=d_p[x];
136       }
137     }
138     int nowMax=maxDepth[circle[0]], nowChosen =
        ↪  maxDepthP[circle[0]];
139     For(i,circleLen-1) {
140       int now=circle[i],len=2*n-2-nowMax-maxDepth[now]-i;
141       pi
          ↪  tmp_p=mp(min(nowChosen,maxDepthP[now]),max(nowChosen,maxDepthP[now]));
142       if (ans>len || ( ans==len && ans_p>tmp_p )) {
143         ans=len;
144         ans_p=tmp_p;
145       }
146       if (nowMax<maxDepth[now]-i || (nowMax==maxDepth[now]-i &&
          ↪  nowChosen > maxDepthP[now])) {
147         nowMax=maxDepth[now]-i;
148         nowChosen = maxDepthP[now];
149       }
150     }
151
152     nowMax=maxDepth[circle[0]], nowChosen = maxDepthP[circle[0]];
153     For(i,circleLen-1) {
154       int now=circle[i],len=2*n-2-nowMax-maxDepth[now]-circleLen+i;
155       pi
          ↪  tmp_p=mp(min(nowChosen,maxDepthP[now]),max(nowChosen,maxDepthP[now]));
156       if (ans>len || ( ans==len && ans_p>tmp_p )) {
157         ans=len;
158         ans_p=tmp_p;
159       }
160       if (nowMax<maxDepth[now]+i || (nowMax==maxDepth[now]-i &&
          ↪  nowChosen > maxDepthP[now])) {
161         nowMax=maxDepth[now]+i;
162         nowChosen = maxDepthP[now];
163       }
164     }
165
166     printf("%d %d %d",ans,ans_p.fi+1,ans_p.se+1);
167   }
168   int main()
169   {
170   //  freopen("hdu5915.in","r",stdin);
171   //  freopen(".out","w",stdout);
172
173     int T=read();
174     For(kcase,T) {
```

```
175       printf("Case #%d: ",kcase);
176       MAIN();
177       puts("");
178     }
179
180     return 0;
181   }
```

”

2.11 la4064_anglesSort

```cpp
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define ForkD(i,k,n) for(int i=n;i>=k;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=Pre[x];p;p=Next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=Next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,127,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define INF (2139062143)
#define F (100000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define eps (1e-9)
#define vi vector<int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Scenario %d:\nThere are %lld sites
    ↪  for making valid tracks\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[i]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
                        For(j,m-1) cout<<a[i][j]<<' ';\
                        cout<<a[i][m]<<endl; \
                        }
#define pi (acos(-1.0))
typedef long long ll;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return (a-b+llabs(a-b)/F*F+F)%F;}
ll gcd(ll a,ll b){if (!b) return a ; return gcd(b,a%b);}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
    int x=0,f=1; char ch=getchar();
    while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
    while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
```

```
44      return x*f;
45  }
46  #define MAXN (4000+10)
47  pair<ll,ll> p[MAXN];
48  double ang(pair<ll,ll> p,pair<ll,ll> p2) { return
    ↪  atan2((double)p2.se-p.se,(double)p2.fi-p.fi); }
49
50  int kcase=0,n;
51  double r[MAXN];
52  int main()
53  {
54  //  freopen("la4064.in","r",stdin);
55      while(cin>>n && n)
56      {
57          For(i,n) cin>>p[i].fi>>p[i].se;
58          ll ans=0;
59          For(i,n) {
60              int m=0;
61              For(j,n)
62                  if (j^i)
63                      Rep(k,2) r[++m]=pi*k*2+ang(p[i],p[j]);
64              sort(r+1,r+m+1);
65              int mv=1,mv2=1;
66              For(j,n-1) {
67                  while (r[mv]<=r[j]+pi/2-eps) ++mv;
68                  while (r[mv2]<r[j]+pi) ++mv2;
69                  int cnt=mv2-mv;
70                  ans += cnt;
71              }
72          }
73          Pr(++kcase,((ll)(n-1)*(n-2)*n/6-ans));
74      }
75  //  {
76  //      freopen("la4064_makedata.in","w",stdout);
77  //      cout<<"100"<<endl;
78  //      For(i,100) {
79  //          int a=rand()%10000,b=rand()%10000;
80  //          if (gcd(a,b)==1) cout<<a<<' '<<b<<endl; else i--;
81  //      }
82  //      cout<<"0\n";
83  //
84  //  }
85      return 0;
86  }
```

”

2.12 middle_KeChiJiuHuaSegmentTree_sum

```cpp
#include<bits/stdc++.h>
using namespace std;
#define For(i,n) for(int i=1;i<=n;i++)
#define Fork(i,k,n) for(int i=k;i<=n;i++)
#define Rep(i,n) for(int i=0;i<n;i++)
#define ForD(i,n) for(int i=n;i;i--)
#define RepD(i,n) for(int i=n;i>=0;i--)
#define Forp(x) for(int p=pre[x];p;p=next[p])
#define Forpiter(x) for(int &p=iter[x];p;p=next[p])
#define Lson (o<<1)
#define Rson ((o<<1)+1)
#define MEM(a) memset(a,0,sizeof(a));
#define MEMI(a) memset(a,0x3f,sizeof(a));
#define MEMi(a) memset(a,128,sizeof(a));
#define MEMx(a,b) memset(a,b,sizeof(a));
#define INF (0x3f3f3f3f)
#define F (1000000007)
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define vi vector<int>
#define pi pair<int,int>
#define SI(a) ((a).size())
#define Pr(kcase,ans) printf("Case #%d: %lld\n",kcase,ans);
#define PRi(a,n) For(i,n-1) cout<<a[i]<<' '; cout<<a[n]<<endl;
#define PRi2D(a,n,m) For(i,n) { \
            For(j,m-1) cout<<a[i][j]<<' ';\
            cout<<a[i][m]<<endl; \
            }
#pragma comment(linker, "/STACK:102400000,102400000")
#define ALL(x) (x).begin(),(x).end()
typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
ll mul(ll a,ll b){return (a*b)%F;}
ll add(ll a,ll b){return (a+b)%F;}
ll sub(ll a,ll b){return ((a-b)%F+F)%F;}
void upd(ll &a,ll b){a=(a%F+b%F)%F;}
int read()
{
  int x=0,f=1; char ch=getchar();
  while(!isdigit(ch)) {if (ch=='-') f=-1; ch=getchar();}
  while(isdigit(ch)) { x=x*10+ch-'0'; ch=getchar();}
```

```
45    return x*f;
46  }
47  #define MAXN (5000000)
48  struct node{
49      int sum,lmax,rmax;
50      node(){};
51      node(int s,int l,int r):sum(s),lmax(l),rmax(r){};
52      friend node operator+(node a,node b) {
53        return
          ↪  node(a.sum+b.sum,max(a.lmax,a.sum+b.lmax),max(b.rmax,b.sum+a.rmax));
54      }
55  }seg[MAXN];
56  int lc[MAXN],rc[MAXN],tot=0,T[MAXN];
57  void build(int l,int r,int &rt){
58      rt=++tot;
59      if(l==r){  seg[rt]=node(1,1,1);  return;}
60      int m=l+r>>1;
61      build(l,m,lc[rt]);  build(m+1,r,rc[rt]);
62      seg[rt]=seg[lc[rt]]+seg[rc[rt]];
63  }
64  void modify(int p,int k,int v,int l,int r,int &rt){
65      rt=++tot;
66      lc[rt]=lc[p];   rc[rt]=rc[p];
67      if(l==r){   seg[rt]=node(v,v,v);    return;}
68      int m=l+r>>1;
69      if(k<=m) modify(lc[p],k,v,l,m,lc[rt]);
70      else        modify(rc[p],k,v,m+1,r,rc[rt]);
71      seg[rt]=seg[lc[rt]]+seg[rc[rt]];
72  }
73  node ask(int L,int R,int l,int r,int rt){
74      if(L>R)  return node(0,0,0);
75      if(L==l&&r==R)  return seg[rt];
76      int m=l+r>>1;
77      if (R<=m) return ask(L,R,l,m,lc[rt]);
78      else if(L>m) return ask(L,R,m+1,r,rc[rt]);
79      else return ask(L,m,l,m,lc[rt])+ask(m+1,R,m+1,r,rc[rt]);
80  }
81  int a[MAXN],q[4],n;
82  pair<int,int> p[MAXN];
83  inline int check(int k){
84      return ask(q[0],q[1],0,n-1,T[k]).rmax+ask(q[1]+1,q[2]-1,0,n-
        ↪  1,T[k]).sum+ask(q[2],q[3],0,n-1,T[k]).lmax  ;
85  }
86  int main()
87  {
88  //  freopen("bzoj3744.in","r",stdin);
```

```
//  freopen(".out","w",stdout);

  n=read();
  Rep(i,n) a[i]=read();
  Rep(i,n) p[i]=mp(a[i],i);
  sort(p,p+n);
  build(0,n-1,T[0]);
  For(i,n-1) modify(T[i-1],p[i-1].se,-1,0,n-1,T[i]);

  int x=0,Q=read();
  while(Q--) {
    Rep(i,4) q[i]=(read()+x)%n;
    sort(q,q+4);
    int l=0,r=n-1,ans=0;
    while(l<=r) {
      int m=(l+r)/2;
      if (check(m)>=0) l=m+1,ans=m;else r=m-1;
    }
    x=p[ans].fi;
    printf("%d\n",x);
  }
  return 0;
}
```

”

2.13 uva465_BigInteger

```cpp
#include <algorithm> // max
#include <cassert>   // assert
#include <cstdio>    // printf,sprintf
#include <cstring>   // strlen
#include <iostream>  // cin,cout
#include <string>    // string 类
#include <vector>    // vector 类
using namespace std;

struct BigInteger {
    typedef unsigned long long LL;

    static const int BASE = 100000000;
    static const int WIDTH = 8;
    vector<int> s;

    BigInteger& clean(){while(!s.back()&&s.size()>1)s.pop_back();
    ↪  return *this;}
    BigInteger(LL num = 0) {*this = num;}
    BigInteger(string s) {*this = s;}
    BigInteger& operator = (long long num) {
        s.clear();
        do {
            s.push_back(num % BASE);
            num /= BASE;
        } while (num > 0);
        return *this;
    }
    BigInteger& operator = (const string& str) {
        s.clear();
        int x, len = (str.length() - 1) / WIDTH + 1;
        for (int i = 0; i < len; i++) {
            int end = str.length() - i*WIDTH;
            int start = max(0, end - WIDTH);
            sscanf(str.substr(start,end-start).c_str(), "%d", &x);
            s.push_back(x);
        }
        return (*this).clean();
    }

    BigInteger operator + (const BigInteger& b) const {
        BigInteger c; c.s.clear();
        for (int i = 0, g = 0; ; i++) {
            if (g == 0 && i >= s.size() && i >= b.s.size()) break;
```

```
            int x = g;
            if (i < s.size()) x += s[i];
            if (i < b.s.size()) x += b.s[i];
            c.s.push_back(x % BASE);
            g = x / BASE;
        }
        return c;
    }
    BigInteger operator - (const BigInteger& b) const {
        assert(b <= *this); // 减数不能大于被减数
        BigInteger c; c.s.clear();
        for (int i = 0, g = 0; ; i++) {
            if (g == 0 && i >= s.size() && i >= b.s.size()) break;
            int x = s[i] + g;
            if (i < b.s.size()) x -= b.s[i];
            if (x < 0) {g = -1; x += BASE;} else g = 0;
            c.s.push_back(x);
        }
        return c.clean();
    }
    BigInteger operator * (const BigInteger& b) const {
        int i, j; LL g;
        vector<LL> v(s.size()+b.s.size(), 0);
        BigInteger c; c.s.clear();
        for(i=0;i<s.size();i++) for(j=0;j<b.s.size();j++)
        ↪  v[i+j]+=LL(s[i])*b.s[j];
        for (i = 0, g = 0; ; i++) {
            if (g ==0 && i >= v.size()) break;
            LL x = v[i] + g;
            c.s.push_back(x % BASE);
            g = x / BASE;
        }
        return c.clean();
    }
    BigInteger operator / (const BigInteger& b) const {
        assert(b > 0);  // 除数必须大于 0
        BigInteger c = *this;       // 商: 主要是让 c.s 和
        ↪  (*this).s 的 vector 一样大
        BigInteger m;               // 余数: 初始化为 0
        for (int i = s.size()-1; i >= 0; i--) {
            m = m*BASE + s[i];
            c.s[i] = bsearch(b, m);
      m -= b*c.s[i];
        }
        return c.clean();
```

```
87      }
88      BigInteger operator % (const BigInteger& b) const { //方法与除
        ↪ 法相同
89          BigInteger c = *this;
90          BigInteger m;
91          for (int i = s.size()-1; i >= 0; i--) {
92              m = m*BASE + s[i];
93              c.s[i] = bsearch(b, m);
94              m -= b*c.s[i];
95          }
96          return m;
97      }
98      // 二分法找出满足 bx<=m 的最大的 x
99      int bsearch(const BigInteger& b, const BigInteger& m) const{
100         int L = 0, R = BASE-1, x;
101         while (1) {
102             x = (L+R)>>1;
103             if (b*x<=m) {if (b*(x+1)>m) return x; else L = x;}
104             else R = x;
105         }
106     }
107     BigInteger& operator += (const BigInteger& b) {*this = *this +
        ↪ b; return *this;}
108     BigInteger& operator -= (const BigInteger& b) {*this = *this -
        ↪ b; return *this;}
109     BigInteger& operator *= (const BigInteger& b) {*this = *this *
        ↪ b; return *this;}
110     BigInteger& operator /= (const BigInteger& b) {*this = *this /
        ↪ b; return *this;}
111     BigInteger& operator %= (const BigInteger& b) {*this = *this %
        ↪ b; return *this;}
112
113     bool operator < (const BigInteger& b) const {
114         if (s.size() != b.s.size()) return s.size() < b.s.size();
115         for (int i = s.size()-1; i >= 0; i--)
116             if (s[i] != b.s[i]) return s[i] < b.s[i];
117         return false;
118     }
119     bool operator >(const BigInteger& b) const{return b < *this;}
120     bool operator<=(const BigInteger& b) const{return !(b <
        ↪ *this);}
121     bool operator>=(const BigInteger& b) const{return !(*this <
        ↪ b);}
122     bool operator!=(const BigInteger& b) const{return b < *this ||
        ↪ *this < b;}
```

```
123     bool operator==(const BigInteger& b) const{return !(b < *this)
        ↪  && !(b > *this);}
124 };
125
126 ostream& operator << (ostream& out, const BigInteger& x) {
127     out << x.s.back();
128     for (int i = x.s.size()-2; i >= 0; i--) {
129         char buf[20];
130         sprintf(buf, "%08d", x.s[i]);
131         for (int j = 0; j < strlen(buf); j++) out << buf[j];
132     }
133     return out;
134 }
135
136 istream& operator >> (istream& in, BigInteger& x) {
137     string s;
138     if (!(in >> s)) return in;
139     x = s;
140     return in;
141 }
142 int main() {
143 //  freopen("uva465.in","r",stdin);
144   BigInteger a,b,L=0x7fffffff,x,y;
145   char c[10];
146
147   while(cin>>x>>c>>y) {
148     cout<<x<<' '<<c<<' '<<y<<endl;
149       if (x > L) puts("first number too big");
150         if (y > L) puts("second number too big");
151         if (c[0] == '+' && x+y > L) puts("result too big");
152         if (c[0] == '*' && x*y > L) puts("result too big");
153   }
154   return 0;
155 }
```

”