Template

iaojnh

September 18, 2018

目录

1 ！绪论

1.1 .vimrc

```
set nu ai ci si mouse=a ts=4 sts=4 sw=4
set mouse=v
nmap<F2> : vs %<.in <CR>
nmap<F3> : !gedit % <CR>
nmap<F8> : !./%< < %<.in <CR>
nmap<F9> : make %<  <CR>

nmap<F5> : !./%< <CR>
nmap<F10> : !g++ % -o %< -O2 -g -Wall <CR>
```

1.2 Hash_Int

```
struct Hash_table {
    static const int V=1000003;
    int fst[V],nxt[V];
    int ctm,ptm[V],T;
    int val[V];
    vector<pair<ll,int>> key;
    void init() { T=0; ctm++; key.clear(); }
    void add(ll s) {
        int S=s%V;
        if (ptm[S]!=ctm) ptm[S]=ctm,fst[S]=-1;
        for (int j=fst[S];j!=-1;j=nxt[j]) if (key[j].fi==s) {
            key[j].se++;
            return;
        }
        nxt[T]=fst[S],fst[S]=T; key.pb(mp(s,1));
        T++;
    }
    int operator [](ll s) {
        int S=s%V;
        if (ptm[S]!=ctm) return 0;
        for (int j=fst[S];j!=-1;j=nxt[j]) if (key[j].fi==s)
            return key[j].se;
        return 0;
    }
}V;
```

1.3 __

```
#include<bits/stdc++.h>
using namespace std;
typedef double db;
typedef long long ll;
typedef vector<int> vi;
typedef pair<int, int> pii;
#define fi first
#define se second
#define mp make_pair
#define pb push_back
#define pw(x) (1ll << (x))
#define sz(x) ((int)(x).size())
#define all(x) (x).begin(),(x).end()
#define rep(i,l,r) for(int i=(l);i<(r);++i)
#define per(i,l,r) for(int i=(r)-1;i>=(l);--i)
#define dd(x) cout << #x << " = " << x << ", "
#define de(x) cout << #x << " = " << x << endl

//------
```

1.4 fast_io

```
struct FastIO {
   static const int S = 1310720;
   int wpos;
   char wbuf[S];
   FastIO() :
         wpos(0) {
   }
   inline int xchar() {
      static char buf[S];
      static int len = 0, pos = 0;
      if (pos == len)
         pos = 0, len = fread(buf, 1, S, stdin);
      if (pos == len)
         return -1;
      return buf[pos++];
   }
   inline int xuint() {
      int c = xchar(), x = 0;
      while (c <= 32)
         c = xchar();
      for (; '0' <= c && c <= '9'; c = xchar())
         x = x * 10 + c - '0';
      return x;
   }
   inline int xint() {
      int s = 1, c = xchar(), x = 0;
      while (c <= 32)
         c = xchar();
      if (c == '-')
         s = -1, c = xchar();
      for (; '0' <= c && c <= '9'; c = xchar())
         x = x * 10 + c - '0';
      return x * s;
   }
   inline void xstring(char *s) {
      int c = xchar();
      while (c <= 32)
         c = xchar();
      for (; c > 32; c = xchar())
         *s++ = c;
      *s = 0;
   }
```

```
   inline void wchar(int x) {
      if (wpos == S)
         fwrite(wbuf, 1, S, stdout), wpos = 0;
      wbuf[wpos++] = x;
   }
   inline void wint(int x) {
      if (x < 0)
         wchar('-'), x = -x;
      char s[24];
      int n = 0;
      while (x || !n)
         s[n++] = '0' + x % 10, x /= 10;
      while (n--)
         wchar(s[n]);
   }
   inline void wstring(const char *s) {
      while (*s)
         wchar(*s++);
   }
   ~FastIO() {
      if (wpos)
         fwrite(wbuf, 1, wpos, stdout), wpos = 0;
   }
} io;
```

2 dp

2.1 数位 dp

```
const int maxn=22;
int dig[maxn];
ll f[maxn]/* [TODO] */;

ll dfs(int pos,/* TODO */,int limit){
    if (pos<0) return /* TODO */;
    if (!limit&&f[pos]/* [TODO] */!=-1) return f[pos]/* [TODO] */;
    LL res=0;
    int last=limit?dig[pos]:9;
    for (int i=0;i<=last;i++){
        res+=dfs(pos-1,/* TODO */,limit&&(i==last));
    }
    if (!limit) f[pos]/* [TODO] */=res;
    return res;
}

ll solve(ll n){
    int len=0;
    while (n){
        dig[len++]=n%10;
        n/=10;
    }
    return dfs(len-1,/* TODO */,1);
}
```

2.2 斜率优化

```
/*
  * dp[i]=Max(dp[j]+(sum[i]-sum[j])^2) (0<=j<i)
  * dp[i]=dp[j]+sum[i]^2+sum[j]^2-2*sum[i]*sum[j] -> dp[j]+sum[j]^2=dp[i]-sum[i]^2+2*sum
  [i]*sum[j]
  * to line y=kx+b
  * y=dp[j]+sum[j]^2,k=2*sum[i],x=sum[j],b=dp[i]-sum[i]^2
*/
ll data[500005];
ll dp[500005]={0},sum[500005]={0};
pii q[500005];
int main()
{
    int n,k;
    while(scanf("%d%d",&n,&k)!=EOF)
    {
        int front=0,rear=-1;
        rep(i,1,n+1)scanf("%lld",&data[i]);
        rep(i,1,n+1)sum[i]=sum[i-1]+data[i];
        rep(i,1,n+1)
        {
            ll x=2*sum[i-1],y=dp[i-1]+sum[i-1]*sum[i-1];
            while(rear-front+1>=2)
            {
                if((y-q[rear].second)*(q[rear].first-q[rear-1].first)<=(q[rear].second-q
                [rear-1].second)*(x-q[rear].first))rear--;
                else break;
            }
            q[++rear]=mp(x,y);
            while(rear-front+1>=2)
            {
                if(sum[i]*(q[front+1].first-q[front].first)>=q[front+1].second-q[front].
                second)front++;
                else break;
            }
            dp[i]=q[front].second-sum[i]*q[front].first+sum[i]*sum[i]+k;
        }
        printf("%lld\n",dp[n]);
    }
    return 0;
}
```

2.3 斯坦纳树

```
// build Steiner Tree (cost,cases),Time Complexity : O(n*3^m)¡£

const int N=55,M=13;
const int mod=1e9+7;
pii f[1<<M][N],g[1<<M][N];//first->value,second->cnt
vector<int> V[N],adt[N];
int tot,vis[N],dis[N],ord[N];
int n,m,q;
pii add(pii a,pii b)
```

```
{
    a.fi+=b.fi;a.se=1LL*a.se*b.se%mod;
    if(a.fi>=n)a=mp(n,0);
    return a;
}
void upd(pii &a,pii b)
{
    if(a.fi>b.fi)a=b;
    else if(a.fi==b.fi)a.se=(a.se+b.se)%mod;
}
ll kpow(ll a,ll b)
{
    ll ans=1;
    while(b)
    {
        if(b&1)ans=ans*a%mod;
        a=a*a%mod;
        b>>=1;
    }
    return ans;
}
int main()
{
    while(scanf("%d%d%d",&n,&q,&m)!=EOF)
    {
        rep(i,0,n)V[i].clear();
        while(q--)
        {
            int a,b;
            scanf("%d%d",&a,&b);a--;b--;
            V[a].pb(b);
            V[b].pb(a);
        }
        //build graph
        rep(i,0,1<<m)rep(j,0,n)f[i][j]=g[i][j]=mp(n,0);
        rep(i,0,n)
        {
            if(i<m)f[1<<i][i]=g[1<<i][i]=mp(0,1);
            else f[0][i]=g[0][i]=mp(0,1);
        }
        //init
        rep(mask,0,1<<m)//S
        {
            rep(i,0,n)
            {
                int cur=(i<m)?(1<<i):0;
                if(cur&&!(cur&mask))continue;
                int rem=mask^cur,lbt=rem&-rem;
                if(rem)
                {
                    for(int mask2=(rem-1)&rem;mask2;mask2=(mask2-1)&rem)
                    {
                        if(mask2&lbt)
                        {
                            int u=mask2|cur,v=(rem^mask2)|cur;//T,S-T
                            assert((u|v)==mask);
                            upd(g[mask][i],add(f[u][i],g[v][i]));//co 2
                        }
                    }
                }
                if(g[mask][i].fi<n)adt[g[mask][i].fi].pb(i);
            }
            tot=0;
            for(int i=0,j=0;i<n;i++)//co 1
            {
                for(auto u:adt[i])
                {
                    if(vis[u]!=mask)
                    {
                        vis[u]=mask;
                        dis[u]=n;
                    }
                    if(dis[u]>i)
                    {
                        dis[u]=i;
                        ord[tot++]=u;
                    }
                }
                adt[i].clear();
                for(;j<tot;j++)
                {
                    int u=ord[j];
                    if(dis[u]>i)break;
                    for(auto v:V[u])
                    {
                        int cur=v<m?1<<v:0;
                        if(cur&&!(mask&cur))continue;
                        if(vis[v]!=mask)
                        {
                            vis[v]=mask;
                            dis[v]=n;
                        }
                        if(dis[v]>dis[u]+1)
                        {
                            dis[v]=dis[u]+1;
                            ord[tot++]=v;
                        }
                    }
                }
            }
            rep(i,0,tot)
            {
                int u=ord[i];
                for(auto v:V[u])
                {
                    int mask2=mask|(v<m?1<<v:0);
                    pii tmp=g[mask][u];
                    ++tmp.fi;
                    upd(f[mask2][v],tmp);
                    upd(g[mask2][v],tmp);
```

```
                }
            }
        }
        pii ans=mp(n,0);
        rep(i,0,n)upd(ans,g[(1<<m)-1][i]);
        if(ans.fi<n)printf("%lld\n",1LL*ans.se*kpow(ans.fi+1,mod-2)%mod);
        else puts("0");
    }
    return 0;
}
```

3 图论

3.1 KM

```
/*
 *  KM
 *  Time Complexity : O(nx*nx*ny)
 *  start with 0
 */
const int N = 310;
const int INF = 0x3f3f3f3f;
int nx,ny;//point number
int g[N][N];//graph
int linker[N],lx[N],ly[N];
int slack[N];
bool visx[N],visy[N];
bool DFS(int x)
{
    visx[x] = true;
    for(int y = 0; y < ny; y++)
    {
        if(visy[y])continue;
        int tmp = lx[x] + ly[y] - g[x][y];
        if(tmp == 0)
        {
            visy[y] = true;
            if(linker[y] == -1 || DFS(linker[y]))
            {
                linker[y] = x;
                return true;
            }
        }
        else if(slack[y] > tmp)
            slack[y] = tmp;
    }
    return false;
}
int KM()
{
    memset(linker,-1,sizeof(linker));
    memset(ly,0,sizeof(ly));
    for(int i = 0;i < nx;i++)
    {
        lx[i] = -INF;
        for(int j = 0;j < ny;j++)
            if(g[i][j] > lx[i])
                lx[i] = g[i][j];
    }
    for(int x = 0;x < nx;x++)
    {
        for(int i = 0;i < ny;i++)
            slack[i] = INF;
        while(true)
        {
            memset(visx,false,sizeof(visx));
            memset(visy,false,sizeof(visy));
            if(DFS(x))break;
            int d = INF;
            for(int i = 0;i < ny;i++)
                if(!visy[i] && d > slack[i])
                    d = slack[i];
            for(int i = 0;i < nx;i++)
                if(visx[i])
                    lx[i] -= d;
            for(int i = 0;i < ny;i++)
            {
                if(visy[i])ly[i] += d;
                else slack[i] -= d;
            }
        }
    }
    int res = 0;
    for(int i = 0;i < ny;i++)
        if(linker[i] != -1)
            res += g[linker[i]][i];
    return res;
}
//HDU 2255
int main()
{
    int n;
    while(scanf("%d",&n) == 1)
    {
        for(int i = 0;i < n;i++)
            for(int j = 0;j < n;j++)
                scanf("%d",&g[i][j]);
        nx = ny = n;
        printf("%d\n",KM());
    }
    return 0;
}
```

3.2 tarjan

```
const int N=100005;
stack<int> s;
vector<int> v[N];
int dfn[N],low[N],instack[N],npoint[N];
```

```
int cnt=0;
int index=0;
void tarjan(int a,int fa)
{
    dfn[a]=low[a]=++index;
    s.push(a);
    instack[a]=1;
    for(int i=0;i<v[a].size();i++)
    {
        int point=v[a][i];
        if(point==fa)continue;
        if(!dfn[point])
        {
            tarjan(point,a);
            low[a]=min(low[a],low[point]);
            /*if(low[point]>=dfn[a]) {
                block.clear();
                int t;
                do {
                    t=s.top();
                    block.pb(t);
                    s.pop();
                }while(t!=point);
                block.pb(a);
                solve();
            }*/
        }
        else if(instack[point])low[a]=min(low[a],dfn[point]);
    }
    if(dfn[a]==low[a])
    {
        cnt++;
        while(s.top()!=a)
        {
            npoint[s.top()]=cnt;
            instack[s.top()]=0;
            s.pop();
        }
        npoint[s.top()]=cnt;
        instack[s.top()]=0;
        s.pop();
    }
}
```

3.3 最大团

```
typedef bool BB[150];
struct Maxclique
{
    const BB *e;int pk,lv;double Tlimit;
    struct ve {int i,d;ve(int i):i(i),d(0){}};
    struct sc {int a,b;sc(     ):a(0),b(0){}};
    typedef vector<ve> ves;ves V;
    typedef vector<int> cc;cc Q,QMAX;
    vector<cc> C;
    vector<sc> S;
    Maxclique(BB *conn,int sz,const double tt=0.025):pk(0),lv(1),Tlimit(tt)
    {
        rep(i,0,sz)V.pb(ve(i));e=conn;
        C.resize(sz+1);
        S.resize(sz+1);
    }
    static bool desc_deg(const ve &a,const ve &b){return a.d>b.d;}
    void ini_col(ves &v){for(int i=sz(V)-1;i>=0;i--)v[i].d=min(i,v[0].d)+1;}
    void set_deg(ves &v){rep(i,0,sz(V)){v[i].d=0;rep(j,0,sz(V))v[i].d+=e[v[i].i][v[j].i
    ];}}
    void deg_sort(ves &R){set_deg(R);sort(R.begin(),R.end(),desc_deg);}
    bool cut1(int pi,cc &va){rep(i,0,sz(va))if(e[pi][va[i]])return true;return false;}
    void cut2(ves &va,ves &vb){rep(i,0,sz(va)-1)if(e[va.back().i][va[i].i])vb.pb(va[i].i
    );}
    void co_sort(ves &R)
    {
        int j=0,maxno=1,min_k=max(sz(QMAX)-sz(Q)+1,1);
        rep(i,1,3)C[i].clear();
        rep(i,0,sz(R))
        {
            int pi=R[i].i,k=1;
            while(cut1(pi,C[k]))k++;
            if(k>maxno)C[(maxno=k)+1].clear();C[k].pb(pi);
            if(k<min_k)R[j++].i=pi;
        }
        if(j>0)R[j-1].d=0;
        rep(k,min_k,maxno+1)rep(i,0,sz(C[k]))R[j].i=C[k][i],R[j++].d=k;
    }
    void exp_dyn(ves &R)
    {
        S[lv].a+=S[lv-1].a-S[lv].b;
        S[lv].b =S[lv-1].a;
        for(;sz(R);Q.pop_back(),R.pop_back())
        {
            if(sz(Q)+R.back().d<=sz(QMAX))return;
            Q.pb(R.back().i);
            ves Rp;cut2(R,Rp);
            if(sz(Rp))
            {
                //if((double)S[lv].a/++pk<Tlimit)deg_sort(Rp);
                co_sort(Rp);S[lv++].a++;
                exp_dyn(Rp);--lv;
            }
            else if(sz(Q)>sz(QMAX))QMAX=Q;
        }
    }
    void mcqdyn(int *mxc,int &sz)
    {
        set_deg(V);sort(V.begin(),V.end(),desc_deg);
        ini_col(V);rep(i,0,sz(V)+1)S[i].a=S[i].b=0;
        exp_dyn(V);for(int i=sz(QMAX)-1;i>=0;i--)mxc[i]=QMAX[i];
        sz=sz(QMAX);
    }
};
```

3.4 最大流

```
const int N=620,M=1000005,LINF=0x3fffffff;
struct MaxFlow {
    // N — V ,  M —  E
    int n, et, dis[N], que[N], cur[N], head[N];
    struct Edge {
        int s, t, v, nxt;
        Edge() {
        }
        Edge(int _s, int _t, int _v, int _nxt) {
            s = _s, t = _t, v = _v, nxt = _nxt;
        }
    } e[M * 2], b[M * 2];
    void undo() {
        for(int i=0;i<et;i++)
            e[i] = b[i];
    }
    void backup() {
        for(int i=0;i<et;i++)
            b[i] = e[i];
    }
    void init(int _n) {
        n = _n, et = 0;
        memset(head, —1, sizeof(head[0]) * n);
    }
    void addEdge(int s, int t, int v) {
        e[et] = Edge(s, t, v, head[s]), head[s] = et++;
        e[et] = Edge(t, s, 0, head[t]), head[t] = et++;
    }
    bool bfs(int S, int T) {
        int qh = 0, qt = 0;
        memset(dis, —1, sizeof(dis[0]) * n);
        dis[S] = 0, que[qt++] = S;
        while (qh < qt)
            for (int i = head[que[qh++]]; ~i; i = e[i].nxt)
                if (e[i].v && !~dis[e[i].t]) {
                    dis[que[qt++] = e[i].t] = 1 + dis[e[i].s];
                    if (e[i].t == T)
                        return true;
                }
        return false;
    }
    ll dinic(int S, int T) {
        int u, qt;
        ll maxflow = 0;
        while (bfs(S, T)) {
            memcpy(cur, head, sizeof(cur[0]) * n);
            u = S, qt = 0;
            while (~cur[S]) {
                if (u == T) {
                    ll flow = LINF;
                    for (int i = qt — 1; i >= 0; ——i)
                        flow = min(flow, (ll) e[que[i]].v);
                    for (int i = qt — 1; i >= 0; ——i) {
                        e[que[i]].v —= flow, e[que[i] ^ 1].v += flow;
                        if (!e[que[i]].v)
                            qt = i;
                    }
                    u = e[que[qt]].s, maxflow += flow;
                } else if (~cur[u] && e[cur[u]].v
                        && dis[u] + 1 == dis[e[cur[u]].t]) {
                    que[qt++] = cur[u];
                    u = e[cur[u]].t;
                } else {
                    while (u != S && !~cur[u])
                        u = e[que[——qt]].s;
                    cur[u] = e[cur[u]].nxt;
                }
            }
        }
        return maxflow;
    }
} G;
```

3.5 最小费用流

```
const ll INF=1e9+7;
const ll LINF=1e15+7;
struct MinCost {
    const static int N = 2e4 + 7;
    const static int M = 1e5 + 7;
    struct Edge {
        int s, t, cap, cost, nxt;
        Edge() {
        }
        Edge(int _s, int _t, int _cap, int _cost, int _nxt) {
            s = _s, t = _t, cap = _cap, cost = _cost, nxt = _nxt;
        }
    } e[M * 2];
    ll dis[N];
    queue<int> que;
    int n, et, pre[N], vis[N], head[N];
    void init(int _n) {
        n = _n, et = 0;
        memset(head, —1, sizeof(head[0]) * n);
    }
    void addEdge(int s, int t, int cap, int cost) {
        e[et] = Edge(s, t, cap, cost, head[s]), head[s] = et++;
        e[et] = Edge(t, s, 0, —cost, head[t]), head[t] = et++;
    }
    bool bfs(int S, int T) {
        rep(i, 0, n)
            pre[i] = —1, dis[i] = LINF, vis[i] = false;
        dis[S] = 0, vis[S] = true, que.push(S);
        while (!que.empty()) {
            int u = que.front();
            assert(dis[u] >= 0);
```

```
            for (int i = head[u]; ~i; i = e[i].nxt) {
                int v = e[i].t;
                if (e[i].cap > 0 && dis[v] > dis[u] + e[i].cost) {
                    pre[v] = i, dis[v] = dis[u] + e[i].cost;
                    if (!vis[v])
                        vis[v] = true, que.push(v);
                }
            }
            vis[u] = false, que.pop();
        }
        return dis[T] != LINF;
    }
    pair<ll, ll> solve(int S, int T) {
        ll maxflow = 0, mincost = 0;
        while (bfs(S, T)) {
            ll flow = INF;
            for (int i = pre[T]; ~i; i = pre[e[i].s])
                flow = min(flow, (ll) e[i].cap);
            for (int i = pre[T]; ~i; i = pre[e[i].s])
                e[i].cap -= flow, e[i ^ 1].cap += flow;
            maxflow += flow, mincost += flow * dis[T];
        }
        return mp(maxflow, mincost);
    }
} G;
```

3.6 树链剖分

```
const int N=50005;
int size[N],deep[N],top[N],father[N],son[N],tid[N],ran[N];
__int64 data[N];
vector<int> V[N];
int total=0;
void dfs1(int u,int fa,int dep)
{
    deep[u]=dep;
    father[u]=fa;
    size[u]=1;
    for(int i=0;i<V[u].size();i++)
    {
        int v=V[u][i];
        if(fa==v)continue;
        dfs1(v,u,dep+1);
        size[u]+=size[v];
        if(son[u]==-1||size[son[u]]<size[v])son[u]=v;
    }
}
void dfs2(int u,int tp)
{
    top[u]=tp;
    tid[u]=++total;
    ran[tid[u]]=u;
    if(son[u]==-1)return;
    dfs2(son[u],tp);
    for(int i=0;i<V[u].size();i++)
    {
        int v=V[u][i];
        if(v!=son[u]&&v!=father[u])dfs2(v,v);
    }
}
//------
__int64 tree[N<<2],lazy[N<<2];
void buildtree(int l,int r,int node)
{
    if(l==r)tree[node]=data[ran[l]];
    else
    {
        int mid=(l+r)>>1;
        buildtree(l,mid,node<<1);
        buildtree(mid+1,r,node<<1|1);
        tree[node]=tree[node<<1]+tree[node<<1|1];
    }
}
void pushdown(int l,int r,int node)
{
    if(lazy[node])
    {
        lazy[node<<1]+=lazy[node];
        lazy[node<<1|1]+=lazy[node];
        int mid=(l+r)>>1;
        tree[node<<1]+=lazy[node]*(mid-l+1);
        tree[node<<1|1]+=lazy[node]*(r-mid);
        lazy[node]=0;
    }
}
void update(int l,int r,int node,int ll,int rr,__int64 num)
{
    if(ll>rr)swap(ll,rr);
    if(ll==l&&rr==r)
    {
        lazy[node]+=num;
        tree[node]+=num*(rr-ll+1);
        return;
    }
    else if(l==r)return;
    int mid=(l+r)>>1;
    pushdown(l,r,node);
    if(rr<=mid)update(l,mid,node<<1,ll,rr,num);
    else if(ll>mid)update(mid+1,r,node<<1|1,ll,rr,num);
    else
    {
        update(l,mid,node<<1,ll,mid,num);
        update(mid+1,r,node<<1|1,mid+1,rr,num);
    }
    tree[node]=tree[node<<1]+tree[node<<1|1];
}
__int64 query(int l,int r,int node,int q)
{
    if(l==r)return tree[node];
    else
```

```
    {
        pushdown(l,r,node);
        int mid=(l+r)>>1;
        if(q<=mid)return query(l,mid,node<<1,q);
        else return query(mid+1,r,node<<1|1,q);
    }
}
void change(int x,int y,int val,int n)
{
    while(top[x]!=top[y])
    {
        if(deep[top[x]]<deep[top[y]])swap(x,y);
        update(1,n,1,tid[top[x]],tid[x],val);
        x=father[top[x]];
    }
    if(deep[x]>deep[y])swap(x,y);
    update(1,n,1,tid[x],tid[y],val);
}
//————————
void init(int n)
{
    memset(deep,0,sizeof(deep));
    memset(size,0,sizeof(size));
    memset(top,0,sizeof(top));
    memset(father,0,sizeof(father));
    memset(son,-1,sizeof(son));
    memset(tid,0,sizeof(tid));
    memset(ran,0,sizeof(ran));
    for(int i=1;i<=n;i++)V[i].clear();
    memset(tree,0,sizeof(tree));
    memset(lazy,0,sizeof(lazy));
    total=0;
}
```

3.7 点分治

```
//k
vector<pii> V[N];
vector<int> dep;
int size,f[N],s[N],root,ans,done[N],d[N];
int n,k;

void getroot(int a,int fa)
{
    s[a]=1;f[a]=0;
    rep(i,0,sz(V[a]))
    {
      int nex=V[a][i].fi;
      if(nex==fa||done[nex])continue;
      getroot(nex,a);
      s[a]+=s[nex];
      f[a]=max(f[a],s[nex]);
    }
    f[a]=max(f[a],size-s[a]);
    if(f[a]<f[root])root=a;
}
void getdep(int a,int fa)
{
   dep.pb(d[a]);;
   s[a]=1;
   rep(i,0,sz(V[a]))
   {
      int nex=V[a][i].fi;
      if(nex==fa||done[nex])continue;
      d[nex]=d[a]+V[a][i].se;
      getdep(nex,a);
      s[a]+=s[nex];
   }
}
int calc(int a,int val)
{
   dep.clear();d[a]=val;
   getdep(a,0);
   sort(all(dep));
   int ret=0;
   for(int l=0,r=sz(dep)-1;l<r;)
   {
      if(dep[l]+dep[r]<=k)ret+=r-l++;
      else r--;
   }
   return ret;
}
void work(int a)
{
   ans+=calc(a,0);
   done[a]=1;
   rep(i,0,sz(V[a]))
   {
      int nex=V[a][i].fi;
      if(done[nex])continue;
      ans-=calc(nex,V[a][i].se);
      f[0]=size=s[a];
      getroot(nex,root=0);
      work(root);
   }
}
int main()
{
   while(scanf("%d%d",&n,&k)!=EOF)
   {
      if(n==0||k==0)break;
      rep(i,1,n+1)V[i].clear(),done[i]=0;
      rep(i,1,n)
      {
         int a,b,c;
         scanf("%d%d%d",&a,&b,&c);
         V[a].pb({b,c});
         V[b].pb({a,c});
      }
      f[0]=size=n;
```

```
      getroot(1,root=0);
      ans=0;
      work(root);
      printf("%d\n",ans);
   }
   return 0;
}
```

3.8 虚树

```
const int N=250005;
const int mod=1e9+7;
vector<pii> V[N];
int Index=0;
int trn[N],rtrn[N];
int Fa[20][N],Dep[N],st[20][N];
int S[N];
int data[N];
void dfs(int a,int fa,int dep)
{
    Fa[0][a]=fa;
    Dep[a]=dep;
    trn[++Index]=a;
    rtrn[a]=Index;
    rep(i,0,sz(V[a]))
    {
        pii nex=V[a][i];
        if(nex.first==fa)continue;
        st[0][nex.first]=nex.second;
        dfs(nex.first,a,dep+1);
    }
}
int lca(int a,int b)
{
    if(Dep[a]<Dep[b])swap(a,b);
    per(i,0,20)if((1<<i)<=Dep[a]-Dep[b])a=Fa[i][a];
    if(a==b)return a;
    per(i,0,20)if(Fa[i][a]!=Fa[i][b])a=Fa[i][a],b=Fa[i][b];
    assert(Fa[0][a]==Fa[0][b]);
    return Fa[0][a];
}
int cmp(int a,int b)
{
    return rtrn[a]<rtrn[b];
}
vector<pii> vec[N];
int exi[N];
ll Dfs(int a,int fa,ll hd)
{
    ll ans=0;
    rep(i,0,sz(vec[a]))
    {
        if(fa==vec[a][i].first)continue;
        ans+=Dfs(vec[a][i].first,a,vec[a][i].second);
    }
    vec[a].clear();
    if(exi[a])return hd;
    return min(ans,hd);
}
int Min(int a,int b)
{
    int ans=0x3f3f3f3f;
    if(Dep[a]<Dep[b])swap(a,b);
    per(i,0,20)if((1<<i)<=Dep[a]-Dep[b])
    {
        ans=min(ans,st[i][a]);
        a=Fa[i][a];
    }
    return ans;
}
int main()
{
    int n;
    scanf("%d",&n);
    memset(st,0x3f,sizeof(st));
    rep(i,1,n)
    {
        int a,b,c;
        scanf("%d%d%d",&a,&b,&c);
        V[a].pb(mp(b,c));
        V[b].pb(mp(a,c));
    }
    dfs(1,0,1);
    rep(i,1,20)rep(j,1,n+1)Fa[i][j]=Fa[i-1][Fa[i-1][j]],st[i][j]=min(st[i-1][j],st[i-1][
    Fa[i-1][j]]);
    int q;
    scanf("%d",&q);
    while(q--)
    {
        int m;
        scanf("%d",&m);
        rep(i,0,m)scanf("%d",&data[i]);
        rep(i,0,m)exi[data[i]]=1;
        sort(data,data+m,cmp);
        int p=0;
        S[p++]=data[0];
        rep(i,1,m)
        {
            int u=data[i];
            int v=S[p-1];
            int Lca=lca(u,v);
            for(;p>1;v=S[p-1])
            {
                int w=S[p-2];
                if(Dep[w]>=Dep[Lca])
                {
                    vec[v].pb(mp(w,Min(v,w)));
                    vec[w].pb(mp(v,Min(v,w)));
                    p--;
                }
```

```
                else break;
            }
            if(Lca!=v)
            {
                vec[Lca].pb(mp(v,Min(Lca,v)));
                vec[v].pb(mp(Lca,Min(Lca,v)));
                p--;
                S[p++]=Lca;
            }
            S[p++]=u;
        }
        while(p>1)
        {
            assert(S[p-1]!=S[p-2]);
            vec[S[p-1]].pb(mp(S[p-2],Min(S[p-1],S[p-2])));
            vec[S[p-2]].pb(mp(S[p-1],Min(S[p-1],S[p-2])));
            p--;
        }
        int root=S[0];
        if(root!=1)
        {
            vec[1].pb(mp(root,Min(1,root)));
            vec[root].pb(mp(1,Min(1,root)));
        }
        printf("%lld\n",Dfs(1,-1,0x3f3f3f3f3f3f3f3f));
        rep(i,0,m)exi[data[i]]=0;
    }
    return 0;
}
```

4 字符串

4.1 AC 自动机

```
struct node{
    node *fail,*s[27];
    int w;
}*head;
char temp[51],str[1000001];
node *getfail(node *p,int k){
    if (p->s[k]!=NULL) return p->s[k];
    else if(p==head) return head;
    else return getfail(p->fail,k);
}
void build_trie(){
    node *root=head;
    node *tep;
    for (int i=0;i<strlen(temp);i++){
        if(root->s[temp[i]-'a']==NULL){
            tep=new node;
            for (int k=0;k<26;k++)tep->s[k]=NULL;
            tep->w=0;
            tep->fail=head;
            root->s[temp[i]-'a']=tep;
        }
        root=root->s[temp[i]-'a'];
        if (i==strlen(temp)-1) root->w+=1;
    }
}
void build_ac(){
    node *root;
    queue<node*> q;
    while (!q.empty())q.pop();
    q.push(head);
    while (!q.empty()){
        root=q.front();
        q.pop();
        for(int i=0;i<26;i++)if(root->s[i]!=NULL){
            q.push(root->s[i]);
            if(root==head)root->s[i]->fail=head;
            else root->s[i]->fail=getfail(root->fail,i);
        }
    }
}
int find(){
    int ans=0;
    int len=strlen(str);
    node* tep;
    node* root=head;
    for(int i=0;i<len;i++){
        while (root->s[str[i]-'a']==NULL&&root!=head)root=root->fail;
        root=(root->s[str[i]-'a']==NULL)?head:root->s[str[i]-'a'];
        tep=root;
        while(tep!=head){
            ans+=tep->w;
            tep->w=0;
            tep=tep->fail;
        }
    }
    return ans;
}
```

4.2 hash

```
const int N=2000005,M=100005,mod1=1e9+7,mod2=1e9+9;
typedef pii hashv;
hashv hs[N],pw[N],base(13331,23333);
hashv operator + (hashv a,hashv b) {
    int c1=a.fi+b.fi,c2=a.se+b.se;
    if (c1>=mod1) c1-=mod1;
    if (c2>=mod2) c2-=mod2;
    return mp(c1,c2);
}
hashv operator - (hashv a,hashv b) {
    int c1=a.fi-b.fi,c2=a.se-b.se;
    if (c1<0) c1+=mod1;
    if (c2<0) c2+=mod2;
    return mp(c1,c2);
}
```

```
hashv operator * (hashv a,hashv b) {
    return mp(1ll*a.fi*b.fi%mod1,1ll*a.se*b.se%mod2);
}
hashv geths(int l,int r) {
    return hs[r]-hs[l-1]*pw[r-l+1];
}
int main() {
    pw[0]=mp(1,1);rep(i,1,N)pw[i]=pw[i-1]*base;
    scanf("%s",s+1);int n=strlen(s+1);
    rep(i,1,n+1)hs[i]=hs[i-1]*base+mp(s[i],s[i]);
}
```

4.3 manacher

```
// str[1..n]
// f[i*2] = len of maxpal center at str[i]
// f[i*2+1] = len of maxpal center at str[i+0.5]
int f[N*2];
void manacher(char* str,int n){
    static char s[N*2];
    rep(i,0,n+1) s[i*2]=str[i], s[i*2+1]='#';
    s[0]='!'; s[2*n+2]='?';
    int a=0,p=0;
    rep(i,1,2*n+2){
        int h=0;
        if(i<=a+p)h=min(f[2*a-i],a+p-i);
        while(s[i+h+1]==s[i-h-1])h++;
        f[i]=h;
        if(i+h>a+p)a=i,p=h;
    }
}
```

4.4 后缀数组

```
const int N=300000+7;
const int INF=0x7fffffff;
int r[N], sa[N], rk[N], het[N];
int wa[N], wb[N], wx[N], wv[N];
bool ise(int *r, int a, int b, int len) {
   return r[a] == r[b] && r[a + len] == r[b + len];
}
void getSa(int n, int m) {
   int *x = wa, *y = wb;
   rep(i, 0, m)
      wx[i] = 0;
   rep(i, 0, n)
      ++wx[x[i] = r[i]];
   rep(i, 1, m)
      wx[i] += wx[i - 1];
   for (int i = n - 1; i >= 0; --i)
      sa[--wx[x[i]]] = i;
   for (int j = 1, p = 0; p < n; j <<= 1, m = p) {
      p = 0;
      rep(i, n - j, n)
         y[p++] = i;
      rep(i, 0, n)
         sa[i] >= j ? y[p++] = sa[i] - j : 0;
      rep(i, 0, m)
         wx[i] = 0;
      rep(i, 0, n)
         ++wx[wv[i] = x[y[i]]];
      rep(i, 1, m)
         wx[i] += wx[i - 1];
      for (int i = n - 1; i >= 0; --i)
         sa[--wx[wv[i]]] = y[i];
      p = 1, swap(x, y);
      x[sa[0]] = 0;
      rep(i, 1, n)
         x[sa[i]] = ise(y, sa[i], sa[i - 1], j) ? p - 1 : p++;
   }
}
void getHeight(int n) { // r[n] != 0
   rep(i, 1, n + 1)
      rk[sa[i]] = i;
   for (int i = 0, k = 0; i < n; het[rk[i++]] = k) {
      k = k > 0 ? k - 1 : 0;
      for (int j = sa[rk[i] - 1]; r[i + k] == r[j + k]; ++k)
         ;
   }
}
/*
r[aa.length()]=0;
getSa(aa.length()+1,30);
getHeight(aa.length());
*/
```

4.5 后缀数组 2

```
namespace Doubling{
     static const int N = 101010;
   // sa[0~n]: 排名第的后缀是以i sa[i] 开头
   // h[1~n]:S[sa[i-1]] 与 S[sa[i]] 的最长公共前缀长度为 h[i]
     int t[N] , wa[N] , wb[N] , sa[N] , h[N];
     void sort(int *x,int *y,int n,int m){
         rep(i,0,m) t[i] = 0;
         rep(i,0,n) t[x[y[i]]]++;
         rep(i,1,m) t[i] += t[i-1];
         per(i,0,n) sa[--t[x[y[i]]]] = y[i];
     }
     bool cmp(int *x,int a,int b,int d){
         return x[a] == x[b] && x[a+d] == x[b+d];
     }
     void da(int *s,int n,int m){
         int *x=wa,*y=wb;
         rep(i,0,n) x[i] = s[i] , y[i] = i;
         sort(x , y , n , m);
         for(int j=1,p=1;p<n;m=p,j<<=1){
             p = 0;rep(i,n-j,n) y[p++] = i;
             rep(i,0,n) if(sa[i] >= j) y[p++] = sa[i] - j;
```

```
            sort(x , y , n , m);
            swap(x , y);p = 1;x[sa[0]] = 0;
            rep(i,1,n) x[sa[i]] = cmp(y,sa[i],sa[i-1],j)?p-1:p++;
        }
    }
    void cal_h(int *s,int n,int *rk){
        int j,k=0;
        for(int i=1;i<=n;++i) rk[sa[i]] = i;
        for(int i=0;i<n;h[rk[i++]] = k)
            for(k&&--k,j=sa[rk[i]-1];s[i+k]==s[j+k];++k);
    }
};
// rank[0~n-1]: 以 i 开头的后缀排名 rank[i]
struct DA{ // [0,n] , in[n] = 0 , n load
    static const int N = 101010;
    int p[18][N] , rk[N] , in[N] , Log[N] , n;
    void Build(){
        Doubling::da(in,n+1,300);
        Doubling::cal_h(in,n,rk);
        Log[0] = -1;for(int i=1;i<=n;++i) Log[i] = Log[i-1] + (i==(i&(-i)));
        for(int i=1;i<=n;++i) p[0][i] = Doubling::h[i];
        for(int j=1;1<<j<=n;++j){
            int lim = n+1-(1<<j);
            for(int i=1;i<=lim;++i)
                p[j][i] = min(p[j-1][i] , p[j-1][i+(1<<j>>1)]);
        }
    }
    // 某两个后缀的最长公共前缀
    int lcp(int a,int b){
        a = rk[a] , b = rk[b];
        if(a > b) swap(a , b);++a;
        int t = Log[b-a+1];
        return min(p[t][a] , p[t][b-(1<<t)+1]);
    }
};
```

4.6 后缀自动机

```
const int N=100005;
struct Node {
    static const int M = 27;
    int mxl;
    Node *pre, *go[M];
    int val;
    void clear(int x) {
        mxl = 0, pre = 0;
        memset(go, 0, sizeof(go));
        val=x;
    }
}*root, *last, *cur, pool[N << 1],*now;
void init() {
    cur = pool;
    root = last = cur++;
    root->clear(-1);
}
void insert(int x,int pos) {
    Node *p = last, *np = cur++;
    np->clear(pos);
    np->mxl = p->mxl + 1;
    while (p && !p->go[x])
        p->go[x] = np, p = p->pre;
    if (!p) {
        np->pre = root;
    } else {
        Node *q = p->go[x];
        if (q->mxl == p->mxl + 1) {
            np->pre = q;
        } else {
            Node *nq = cur++;
            nq->clear(q->val);
            nq->pre = q->pre, nq->mxl = p->mxl + 1;
            memcpy(nq->go, q->go, sizeof(nq->go));
            np->pre = q->pre = nq;
            while (p && p->go[x] == q)
                p->go[x] = nq, p = p->pre;
        }
    }
    last = np;
}
char str[N];
int main()//2015 □□□□°G
{
    int T;
    scanf("%d",&T);
    rep(cas,0,T)
    {
        printf("Case #%d:\n",cas+1);
        init();
        scanf("%s",str);
        int p=0,n=strlen(str);
        while(p!=n)
        {
            now=root;
            int dis=0;
            while(p<n&&now->go[str[p]-'a']!=0)
            {
                now=now->go[str[p]-'a'];
                dis++;
                insert(str[p]-'a',p);
                p++;
            }
            if(dis==0)
            {
                insert(str[p]-'a',p);
                printf("-1 %d\n",str[p++]);
            }
            else printf("%d %d\n",dis,now->val-dis+1);
        }
    }
    return 0;
```

```
}
```

4.7 回文树

```
const int N = 1e6 + 7;
char s[N];
int n;//length of s
struct Palindromic_Tree
{
   static const int M=27,N=::N+2;
   int size[N],len;
   struct Node {
      int son[M];
      int ff,len;
   }t[N];
   int last,tot;
   void init() {
       last=0,tot=1;
       len=n;
       memset(t,0,sizeof(Node)*(len+5));
       memset(size,0,sizeof(int)*(len+5));
       t[tot++].len=-1;
       t[0].ff=t[1].ff=1;
   }
   void build(char *s) {
      rep(i,0,len) {
         int c=s[i]-'a';
         int p=last;
         while(s[i-t[p].len-1]!=s[i])p=t[p].ff;
         if(!t[p].son[c]) {
            int v=tot++,k=t[p].ff;
            t[v].len=t[p].len+2;
            while(s[i-t[k].len-1]!=s[i])k=t[k].ff;
            t[v].ff=t[k].son[c];
            t[p].son[c]=v;
         }
         last=t[p].son[c];
         size[t[p].son[c]]++;
      }
   }
};
```

5 数学

5.1 fft

```
const double Pi=acos(-1);
const ll mod=998244353;
const int N=100005*4;
int inc(int &a,int b)
{
    a+=b;
    if(a>mod)a-=mod;
}
struct C {
    double r, i;
    C() {
        r = i = 0;
    }
    C(double _r, double _i) {
        r = _r, i = _i;
    }
    C operator+(const C &p) const {
        return C(r + p.r, i + p.i);
    }
    C operator-(const C &p) const {
        return C(r - p.r, i - p.i);
    }
    C operator*(const C &p) const {
        return C(r * p.r - i * p.i, r * p.i + i * p.r);
    }
};
inline C conj(const C &p) {
    return C(p.r, -p.i);
}
C w[N], A[N], B[N], dfa[N], dfb[N], dfc[N], dfd[N];
int L, bitrev[N];
void init(int len) {
    L = 0;
    while (1 << L <= len)
        ++L;
    int n = 1 << L;
    rep(i, 0, n)
        bitrev[i] = (bitrev[i >> 1] >> 1) | ((i & 1) << (L - 1));
    rep(i, 0, n)
        w[i] = C(cos(2 * Pi * i / n), sin(2 * Pi * i / n));
}
void fft(C a[], const int &n) {
    rep(i, 0, n)
        if (i < bitrev[i])
            swap(a[i], a[bitrev[i]]);
    for (int i = 2, d = n >> 1; i <= n; i <<= 1, d >>= 1)
        for (int j = 0; j < n; j += i) {
            C *l = a + j, *r = a + j + (i >> 1), *p = w;
            for (int k = 0; k < (i >> 1); ++k) {
                C tmp = (*r) * (*p);
                *r = *l - tmp, *l = *l + tmp;
                ++l, ++r, p += d;
            }
        }
}
vector<int> gao(const vector<int> &a, const vector<int> &b) {
    init(sz(a) + sz(b));
    int n = 1 << L;
    rep(i, 0, n)
        A[i] = B[i] = C(0, 0);
    rep(i, 0, sz(a))
        A[i] = C(a[i] & 32767, a[i] >> 15);
```

```
    rep(i, 0, sz(b))
        B[i] = C(b[i] & 32767, b[i] >> 15);
    fft(A, n), fft(B, n);
    rep(i, 0, n) {
        int j = (n - i) & (n - 1);
        static C da, db, dc, dd;
        da = (A[i] + conj(A[j])) * C(0.5, 0);
        db = (A[i] - conj(A[j])) * C(0, -0.5);
        dc = (B[i] + conj(B[j])) * C(0.5, 0);
        dd = (B[i] - conj(B[j])) * C(0, -0.5);
        dfa[j] = da * dc, dfb[j] = da * dd;
        dfc[j] = db * dc, dfd[j] = db * dd;
    }
    rep(i, 0, n) {
        A[i] = dfa[i] + dfb[i] * C(0, 1);
        B[i] = dfc[i] + dfd[i] * C(0, 1);
    }
    fft(A, n), fft(B, n);
    vector<int> ret(n, 0);
    rep(i, 0, n) {
        ll da = (ll) (A[i].r / n + 0.5) % mod;
        ll db = (ll) (A[i].i / n + 0.5) % mod;
        ll dc = (ll) (B[i].r / n + 0.5) % mod;
        ll dd = (ll) (B[i].i / n + 0.5) % mod;
        inc(ret[i], (da + ((db + dc) << 15) + (dd << 30)) % mod);
    }
    return ret;
}
// no mod
const int N = 500005*4;
const double Pi=acos(-1);
struct C {
   double r, i;
   C() {
      r = i = 0;
   }
   C(double _r, double _i) {
      r = _r, i = _i;
   }
   C operator+(const C &p) const {
      return C(r + p.r, i + p.i);
   }
   C operator-(const C &p) const {
      return C(r - p.r, i - p.i);
   }
   C operator*(const C &p) const {
      return C(r * p.r - i * p.i, r * p.i + i * p.r);
   }
};
void fft(C x[], int n, int rev) {
   int i, j, k, t;
   for (i = 1; i < n; ++i) {
      for (j = 0, k = n >> 1, t = i; k; k >>= 1, t >>= 1)
         j = j << 1 | (t & 1);
      if (i < j)
         swap(x[i], x[j]);
   }
   int s, ds;
   for (s = 2, ds = 1; s <= n; ds = s, s <<= 1) {
      C w = C(1, 0), t;
      C wn = C(cos(2.0 * rev * Pi / s), sin(2.0 * rev * Pi / s));
      for (k = 0; k < ds; ++k, w = w * wn)
         for (i = k; i < n; i += s) {
            t = w * x[i + ds];
            x[i + ds] = x[i] - t;
            x[i] = x[i] + t;
         }
   }
   if (rev == -1)
      for (i = 0; i < n; ++i)
         x[i].r /= n;
}
```

5.2 伯努利数

```
/* 1^k+2^k+3^k+¡-¡-+n^k
 * O(k^2)
*/
struct Bell
{
    static const int N=::N;
    static const int MOD=::mod;
    int C[N][N],B[N];
    int pow_mod(int x,int k)
    {
        int ans=1;
        while(k)
        {
            if(k&1)ans=1LL*ans*x%mod;
            x=1LL*x*x%mod;
            k>>=1;
        }
        return ans;
    }
    void init()
    {
        rep(i,0,N)
        {
            C[i][0]=C[i][i]=1;
            rep(j,1,i)C[i][j]=(C[i-1][j-1]+C[i-1][j])%mod;
        }
        B[0]=1;
        rep(i,1,N)
        {
            B[i]=0;
            rep(j,0,i)B[i]=(1LL*B[i]-1LL*C[i+1][j]*B[j]%MOD+MOD)%MOD;
            B[i]=1LL*B[i]*pow_mod(C[i+1][i],MOD-2)%MOD;
        }
    }
    int cal(int n,int k)
```

```
    {
        int ans=pow_mod(k+1,MOD-2),sum=0;
        rep(i,0,k+1)sum=(sum+1LL*C[k+1][i]*B[i]%MOD*pow_mod(n,k+1-i)%MOD)%MOD;
        return 1LL*ans*sum%MOD;
    }
}B;
```

5.3 佩尔方程求解

```
// 不定方程 x^2-dy的所有正整数解^2=1
bool PQA(ll D, ll &p, ll &q) //最小解
{
    ll d = sqrt(D);
    if ((d + 1) * (d + 1) == D) return false;
    if (d * d == D)             return false;
    if ((d - 1) * (d - 1) == D) return false;
    ll u = 0, v = 1, a = int(sqrt(D)), a0 = a, lastp = 1, lastq = 0;
    p = a, q = 1;
    do {
        u = a * v - u;
        v = (D - u * u) / v;
        a = (a0 + u) / v;
        ll thisp = p, thisq = q;
        p = a * p + lastp;
        q = a * q + lastq;
        lastp = thisp;
        lastq = thisq;
    } while ((v != 1 && a <= a0));
    p = lastp;
    q = lastq;
    if (p * p - D * q * q == -1) {
        p = lastp * lastp + D * lastq * lastq;
        q = 2 * lastp * lastq;
    }
    return true;
}
/* 求出最小的(x1,y1)后，使用如下迭代式求通解
 * x_{n}=x_{1}*x_{n-1}+dy_{1}*y_{n-1}
 * y_{n}=y_{1}*x_{n-1}+ x_{1}*y_{n-1}
*/
```

5.4 小数高斯消元

```
bool used[N];
double x[N], a[N][N];
int gauss(int n, int m) {
   int row, col;
   for (row = col = 0; row < n && col < m; ++row, ++col) {
      int mxr = row;
      rep(i, row + 1, n)
         if (fabs(a[i][col]) > fabs(a[mxr][col]))
            mxr = i;
      if (fabs(a[mxr][col]) < EPS) {
         --row;
         continue;
      }
      if (mxr != row)
         swap(a[row], a[mxr]);
      rep(i, 0, n)
         if (i != row && fabs(a[i][col]) > EPS)
            for (int j = m; j >= col; --j)
               a[i][j] -= a[row][j] * a[i][col] / a[row][col];
   }
   if (row == n && n == m) {
      rep(i, 0, n)
         x[i] = a[i][m] / a[i][i];
//    rep(i, 0, n)
//       printf("x[%d] = %lf\n", i, x[i]);
      return 1;
   }
   rep(i, row + 1, n)
      if (a[i][m] > EPS)
         return -1; // No Solution
   return 0;
   // free_num = m - row
   memset(used, false, sizeof(used[0]) * m);
   rep(i, 0, row)
   {
      int cnt = 0, ind = -1;
      rep(j, 0, m)
         if (fabs(a[i][j]) > EPS && !used[j])
            ++cnt, ind = j;
      if (cnt == 1)
         x[ind] = a[i][m] / a[i][ind], used[ind] = true;
   }
// rep(i, 0, m)
//    if (used[i])
//       printf("x[%d] = %lf\n", i, x[i] + EPS);
//    else
//       printf("x[%d] = not determined\n", i);
}
```

5.5 拉格朗日差值

```
namespace polysum {
    const int D=101000;
    ll a[D],f[D],g[D],p[D],p1[D],p2[D],b[D],h[D][2],C[D];
    void init(int M) {
        f[0]=f[1]=g[0]=g[1]=1;
        rep(i,2,M+5) f[i]=f[i-1]*i%mod;
        g[M+4]=powmod(f[M+4],mod-2);
        per(i,1,M+4) g[i]=g[i+1]*(i+1)%mod;
    }
    ll calcn(int d,ll *a,ll n) { // a[0].. a[d]  a[n]
        if (n<=d) return a[n];
        p1[0]=p2[0]=1;
        rep(i,0,d+1) {
            ll t=(n-i+mod)%mod;
            p1[i+1]=p1[i]*t%mod;
```

```
        }
        rep(i,0,d+1) {
            ll t=(n-d+i+mod)%mod;
            p2[i+1]=p2[i]*t%mod;
        }
        ll ans=0;
        rep(i,0,d+1) {
            ll t=g[i]*g[d-i]%mod*p1[i]%mod*p2[d-i]%mod*a[i]%mod;
            if ((d-i)&1) ans=(ans-t+mod)%mod;
            else ans=(ans+t)%mod;
        }
        return ans;
    }
    ll polysum(ll n,ll *a,ll m) { // a[0].. a[m] \sum_{i=0}^{n-1} a[i]
        a[m+1]=calcn(m,a,m+1);
        rep(i,1,m+2) a[i]=(a[i-1]+a[i])%mod;
        return calcn(m+1,a,n-1);
    }
    ll qpolysum(ll R,ll n,ll *a,ll m) { // a[0].. a[m] \sum_{i=0}^{n-1} a[i]*R^i
        if (R==1) return polysum(n,a,m);
        a[m+1]=calcn(m,a,m+1);
        ll r=powmod(R,mod-2),p3=0,p4=0,c,ans;
        h[0][0]=0;h[0][1]=1;
        rep(i,1,m+2) {
            h[i][0]=(h[i-1][0]+a[i-1])*r%mod;
            h[i][1]=h[i-1][1]*r%mod;
        }
        rep(i,0,m+2) {
            ll t=g[i]*g[m+1-i]%mod;
            if (i&1) p3=((p3-h[i][0]*t)%mod+mod)%mod,p4=((p4-h[i][1]*t)%mod+mod)%mod;
            else p3=(p3+h[i][0]*t)%mod,p4=(p4+h[i][1]*t)%mod;
        }
        c=powmod(p4,mod-2)*(mod-p3)%mod;
        rep(i,0,m+2) h[i][0]=(h[i][0]+h[i][1]*c)%mod;
        rep(i,0,m+2) C[i]=h[i][0];
        ans=(calcn(m,C,n)*powmod(R,n)-c)%mod;
        if (ans<0) ans+=mod;
        return ans;
    }
}
```

5.6 整数高斯消元

```
struct Gauss {
   static const int N = 305, P = 1e9 + 7;
   int a[N][N];
   int kpow(int a, int b) {
      ll r = 1;
      for (; b; b >>= 1, a = a * a % P)
         if (b & 1)
            r = r * a % P;
      return r;
   }
   int solve(int n, int m) {
      int i = 0, x = 0;
      for (; i < n && x < m; ++i, ++x) {
         int r = i;
         while (r < n && !a[r][x])
            ++r;
         if (r < n) {
            if (r != i)
               rep(j, 0, m + 1)
                  swap(a[r][j], a[i][j]);
//          rep(j, 0, n)
//             if (i != j && a[j][x])
//                for (int k = m; k >= x; --k)
//                   a[j][k] = (a[j][k]
//                         - 1ll * a[i][k] * a[j][x] % P
//                               * kpow(a[i][x], P - 2) % P + P) % P;
            rep(j, 0, n)
               if (i != j && a[j][x])
                  for (int k = m; k >= x; --k)
                     a[j][k] -= a[i][k] * a[j][x] / a[i][x];
         } else
            --i;
      }
      rep(k, i, n)
         if (a[k][m])
            return -1;
      return m - i;
   }
   void output(int n, int m) {
      rep(i, 0, n)
      {
         rep(j, 0, m)
            printf("%d ", a[i][j]);
         puts("");
      }
   }
} G;
```

5.7 线性基

```
struct Gauss {
   int cnt;
   ll st, lb[25];
   void init() {
      cnt = 0, st = 0;
      rep(i, 0, 61) lb[i] = 0;
   }
   void add(ll w) {
      st |= w;
      for (int i = 60; i >=0; --i) {
         if ((w >> i & 1) && !lb[i]) {
            lb[i] = w, ++cnt;
            break;
         }
         w = min(w, w ^ lb[i]);
      }
   }
```

```
}
```

6 数据结构

6.1 LCT

```
struct LCT {
    bool rt[N], rev[N];
    int n, fa[N], que[N], ch[N][2];
    // custom information
    void init(int _n) {
        n = _n;
        rep(i, 0, n) {
            rt[i] = true, rev[i] = false;
            fa[i] = ch[i][0] = ch[i][1] = 0;
        }
    }
    void reverse(int x) {
        rev[x] = !rev[x], swap(ch[x][0], ch[x][1]);
    }
    void up(int x) {
    }
    void down(int x) {
        if (rev[x])
            rev[x] = 0, reverse(ch[x][0]), reverse(ch[x][1]);
    }
    void rotate(int x) {
        int y = fa[x], k = (ch[y][0] == x);
        ch[y][!k] = ch[x][k];
        fa[ch[x][k]] = y, fa[x] = fa[y];
        fa[ch[x][k] = y] = x;
        if (rt[y])
            rt[y] = false, rt[x] = true;
        else
            ch[fa[x]][ch[fa[x]][1] == y] = x;
        up(y);
    }
    void update(int x) {
        int top = 0;
        que[top++] = x;
        while (!rt[x]) x = fa[x], que[top++] = x;
        while (top) down(que[--top]);
    }
    void splay(int x) {
        update(x);
        while (!rt[x]) {
            int y = fa[x], z = fa[y];
            if (!rt[y])
                (ch[z][1] == y) == (ch[y][1] == x) ? rotate(y) : rotate(x);
            rotate(x);
        }
        up(x);
    }
    // x->root be a preferred path, like heavy chain
    void access(int x) {
        for (int y = 0; x; y = x, x = fa[x]) {
            splay(x);
            rt[ch[x][1]] = true;
            rt[ch[x][1] = y] = false;
            up(x);
        }
    }
    int getRoot(int x) {
        access(x), splay(x);
        while (ch[x][0]) x = ch[x][0];
        return x;
    }
    // make x be tree root
    void makeRoot(int x) {
        access(x), splay(x), reverse(x);
    }
    // be sure x,y not in one tree
    void addEdge(int x, int y) {
        makeRoot(x), fa[x] = y;
    }
    // delete edge between(x, parent_x)
    void cut(int x) {
        access(x), splay(x);
        if (ch[x][0])
            fa[ch[x][0]] = fa[x], rt[ch[x][0]] = true;
        fa[x] = ch[x][0] = 0;
    }
    void delEdge(int x, int y) {
        // if y is father of x, can remove operation makeRoot(y)
        makeRoot(y), cut(x);
    }
    void delNode(int x) {
        splay(x);
        rep(i, 0, 2) {
            if (!ch[x][i]) continue;
            fa[ch[x][i]] = fa[x], rt[ch[x][i]] = true;
            fa[x] = ch[x][i] = 0;
        }
    }
} lct;
```

6.2 splay

```
#define key tree[tree[root].ch[1]].ch[0]
struct node
{
    int data,ch[2],fa,size;//□□□□□±
    int add,rev;//□□lazy
    bitset<256> B;
}tree[100005];
int data[100005];
int root;
//————————data————————
void update(int x)
```

```
{
    tree[x].size=1+tree[tree[x].ch[0]].size+tree[tree[x].ch[1]].size;
    tree[x].B=tree[tree[x].ch[0]].B|tree[tree[x].ch[1]].B;
    tree[x].B[tree[x].data]=1;
}
int buildtree(int l,int r,int fa)
{
    if(l>r)return 0;
    int x=(l+r)>>1;
    tree[x].data=data[x];
    tree[x].fa=fa;
    tree[x].add=tree[x].rev=0;
    tree[x].ch[0]=buildtree(l,x-1,x);
    tree[x].ch[1]=buildtree(x+1,r,x);
    update(x);
    return x;
}
void add(int x,int val)
{
    if(!x)return;
    tree[x].data=(tree[x].data+val)%256;
    tree[x].add=(tree[x].add+val)%256;
    tree[x].B=(tree[x].B<<val)|(tree[x].B>>(256-val));
}
void rev(int x)
{
    swap(tree[x].ch[0],tree[x].ch[1]);
    tree[x].rev^=1;
}
void pushdown(int x)
{
    if(tree[x].add)
    {
        add(tree[x].ch[0],tree[x].add);
        add(tree[x].ch[1],tree[x].add);
        tree[x].add=0;
    }
    if(tree[x].rev)
    {
        rev(tree[x].ch[0]);
        rev(tree[x].ch[1]);
        tree[x].rev=0;
    }
}
void rot(int x)
{
    int y=tree[x].fa;
    pushdown(y);
    pushdown(x);
    if (root==y)root=x;
    int f=tree[y].ch[0]==x;
    tree[y].ch[!f]=tree[x].ch[f];
    if (tree[x].ch[f]) tree[tree[x].ch[f]].fa=y;
    tree[x].fa=tree[y].fa;
    if(tree[y].fa) tree[tree[y].fa].ch[tree[tree[y].fa].ch[1]==y]=x;
    tree[x].ch[f]=y;
    tree[y].fa=x;
    update(y);
}
void splay(int x,int g)
{
    if(tree[x].fa==g)pushdown(x);
    else
    {
        while(tree[x].fa!=g)rot(x);
        update(x);
    }
}
void rto(int k,int g)
{
    int x=root;
    while(1)
    {
        pushdown(x);
        if(tree[tree[x].ch[0]].size+1==k)break;
        if(k<=tree[tree[x].ch[0]].size)x=tree[x].ch[0];
        else
        {
            k-=tree[tree[x].ch[0]].size+1;
            x=tree[x].ch[1];
        }
    }
    splay(x,g);
}
//————————function————————
int main()
{
    int T;
    scanf("%d",&T);
    while(T--)
    {
        int n,m,q;
        scanf("%d%d%d",&n,&m,&q);
        rep(i,2,n*m+2)scanf("%d",&data[i]);
        root=buildtree(1,n*m+2,0);
        while(q--)
        {
            int co;
            scanf("%d",&co);
            switch(co)
            {
                case 1:
                {
                    int a,b;
                    scanf("%d%d",&a,&b);
                    int qu=(a-1)*m+b+1;
                    rto(qu-1,0);
                    rto(qu+1,root);
                    printf("%d\n",tree[key].data);
                    break;
```

```
                }
                case 2:
                {
                    int a,b,c;
                    scanf("%d%d%d",&a,&b,&c);
                    int l=(a-1)*m+b+1;
                    int r=(a-1)*m+c+1;
                    rto(l-1,0);
                    rto(r+1,root);
                    printf("%u\n",tree[key].B.count());
                    break;
                }
                case 3:
                {
                    int a,b,c,val;
                    scanf("%d%d%d%d",&a,&b,&c,&val);
                    val%=256;
                    int l=(a-1)*m+b+1;
                    int r=(a-1)*m+c+1;
                    rto(l-1,0);
                    rto(r+1,root);
                    add(key,val);
                    break;
                }
                case 4:
                {
                    int a,b,c;
                    scanf("%d%d%d",&a,&b,&c);
                    int l=(a-1)*m+b+1;
                    int r=(a-1)*m+c+1;
                    rto(l-1,0);
                    rto(r+1,root);
                    rev(key);
                    break;
                }
            }
        }
    }
    return 0;
}
```

6.3 主席树

```
const int N=100005;
const int NN=2000005;
int data[N]={0};
int pos[N]={0};
vector<int> V;
int s=0;
int l[NN]={0},r[NN]={0},c[NN]={0},total=0,Null=1;
int treenode[N]={0};
void buildtree(int &x, int ll,int rr)
{
    x=++total;
    if(ll<rr)
    {
        buildtree(l[x],ll,(ll+rr)>>1);
      buildtree(r[x],((ll+rr)>>1)+1,rr);
    }
}
int in(int y,int num)
{
    int root=++total;
    int x=root;
    int ll=1,rr=s;
    c[x]=c[y]+1;
    while(ll<rr)
    {
        int mid=(ll+rr)>>1;
        if(num<=mid)
        {
            l[x]=++total;
            r[x]=r[y];
            x=l[x];y=l[y];
            rr=mid;
        }
        else
        {
            l[x]=l[y];
            r[x]=++total;
            x=r[x];y=r[y];
            ll=mid+1;
        }
        c[x]=c[y]+1;
    }
    return root;
}
int query(int a,int b,int k)
{
    int x=treenode[a-1],y=treenode[b];
    int ll=1,rr=s;
    while(ll<rr)
    {
        int mid=(ll+rr)>>1;
        int lnum=c[l[y]]-c[l[x]];
        if(lnum>=k)
        {
            x=l[x];y=l[y];
            rr=mid;
        }
        else
        {
            k-=lnum;
            x=r[x];y=r[y];
            ll=mid+1;
        }
    }
    return ll;
}
int main()
```

```
{
    int n,m;
    scanf("%d%d",&n,&m);
    for(int i=1;i<=n;i++)scanf("%d",&data[i]);
//——————————
    for(int i=1;i<=n;++i)V.push_back(data[i]);
    sort(V.begin(),V.end());
    V.erase(unique(V.begin(),V.end()),V.end());
    for(int i=1;i<=n;++i)pos[i]=lower_bound(V.begin(),V.end(),data[i])-V.begin()+1;
//——————————
    s=V.size();
    buildtree(Null,1,s);
    treenode[0]=1;
    for(int i=1;i<=n;i++)treenode[i]=in(treenode[i-1],pos[i]);
//——————————
    while(m--)
    {
        int a,b,c;
        scanf("%d%d%d",&a,&b,&c);
        printf("%d\n",V[query(a,b,c)-1]);
    }
    return 0;
}
```

6.4 线段树

```
const int N=50005;
int data[N];
long long tree[N<<2],lazy[N<<2];
void buildtree(int l,int r,int node)
{
    if(l==r)tree[node]=data[l];
    else
    {
        int mid=(l+r)>>1;
        buildtree(l,mid,node<<1);
        buildtree(mid+1,r,node<<1|1);
        tree[node]=tree[node<<1]+tree[node<<1|1];
    }
}
void pushdown(int l,int r,int node)
{
    if(lazy[node])
    {
        lazy[node<<1]+=lazy[node];
        lazy[node<<1|1]+=lazy[node];
        int mid=(l+r)>>1;
        tree[node<<1]+=lazy[node]*(mid-l+1);
        tree[node<<1|1]+=lazy[node]*(r-mid);
        lazy[node]=0;
    }
}
void update(int l,int r,int node,int _l,int _r,ll num)
{
    if(_l==l&&_r==r)
    {
        lazy[node]+=num;
        tree[node]+=num*(_r-_l+1);
        return;
    }
    else if(l==r)return;
    int mid=(l+r)>>1;
    pushdown(l,r,node);
    if(_r<=mid)update(l,mid,node<<1,_l,_r,num);
    else if(_l>mid)update(mid+1,r,node<<1|1,_l,_r,num);
    else
    {
        update(l,mid,node<<1,_l,mid,num);
        update(mid+1,r,node<<1|1,mid+1,_r,num);
    }
    tree[node]=tree[node<<1]+tree[node<<1|1];
}
long long query(int l,int r,int node,int _l,int _r)
{
    if(l!=r)pushdown(l,r,node);
    int mid=(l+r)>>1;
    if(_l>r||_r<l)return 0;
    else if(_l<=l&&_r>=r)return tree[node];
    else return query(l,mid,node<<1,_l,_r)+query(mid+1,r,node<<1|1,_l,_r);
}
```