Contents

1 费用流

```
const int inf=~0u>>1;
int n,m;
int S,T;
int ans;
struct node{
    int to,f,v,next;
} w[1000001];
int h[1001],cnt=1;
void add(int x,int y,int f,int v) {
    w[++cnt]=(node){y,f,v,h[x]};
    h[x]=cnt;
    w[++cnt]=(node){x,0,-v,h[y]};
    h[y]=cnt;
}
queue<int> q;
int d[1001];
bool vst[1001];
int fa[1001];
bool Spfa() {
    int i,j,v;
    while (!q.empty()) q.pop();
    memset(vst,0,sizeof vst);
    for (i=1; i<=T; i++) d[i]=-inf;
    memset(fa,0,sizeof fa);
    d[S]=0;
    vst[S]=1;
    q.push(S);
    while (!q.empty()) {
        i=q.front();
        q.pop();
        vst[i]=0;
        for (v=h[i]; v; v=w[v].next) {
            j=w[v].to;
            if (w[v].f&&d[i]+w[v].v>d[j]) {
                d[j]=d[i]+w[v].v;
                fa[j]=v;
                if (!vst[j]) {
                    vst[j]=1;
                    q.push(j);
                }
            }
        }
    }
    return d[T]!=-inf;
}
void Adjust() {
```

```
    int i,v,f=inf;
    i=T;
    while (i!=S) {
        v=fa[i];
        f=min(f,w[v].f);
        i=w[v^1].to;
    }
    ans+=f*d[T];
    i=T;
    while (i!=S) {
        v=fa[i];
        w[v].f-=f;
        w[v^1].f+=f;
        i=w[v^1].to;
    }
}
int main() {
  while (Spfa()) Adjust();
}
```

2 hdu2665(可持久线段树)

```
#include<iostream>
#include<cstdio>
#include<algorithm>
#define mp make_pair
#define st first
#define nd second
using namespace std;
const int inf=~0u>>1;
int n,m;
int ans;
int a[100001];
pair<int,int> b[100001];
int f[100001];
struct ST{
    int ls,rs,v;
} T[2000001];
int tot;
int h[100001];
void build(int u,int l,int r) {
    int mid;
    if (l==r) return;
    mid=l+r>>1;
    T[u].ls=++tot;
    T[u].rs=++tot;
    T[u].v=0;
    build(T[u].ls,l,mid);
    build(T[u].rs,mid+1,r);
}
int add(int u,int l,int r,int k) {
    int mid;
    T[++tot]=T[u];
    u=tot;
    T[u].v++;
    if (l==r) return u;
    mid=l+r>>1;
    if (k<=mid)
        T[u].ls=add(T[u].ls,l,mid,k);
    else T[u].rs=add(T[u].rs,mid+1,r,k);
    return u;
}
int ask(int u,int v,int l,int r,int k) {
    int mid,t;
    if (l==r) return l;
    mid=l+r>>1;
    t=T[T[v].ls].v-T[T[u].ls].v;
    if (t>=k) return ask(T[u].ls,T[v].ls,l,mid,k);
    else return ask(T[u].rs,T[v].rs,mid+1,r,k-t);
}
int main() {
    int __,i,j,l,r,v;
    scanf("%d",&__);
    while (__--) {
        scanf("%d%d",&n,&m);
        for (i=1; i<=n; i++) {
            scanf("%d",&v);
            b[i]=mp(v,i);
        }
        sort(b+1,b+n+1);
        b[0]=mp(inf,inf);
        for (i=1,j=0; i<=n; i++) {
            if (b[i]!=b[i-1]) {
                j++;
                f[j]=b[i].st;
            }
            a[b[i].nd]=j;
        }
        tot=1;
        build(1,1,n);
        h[0]=1;
        for (i=1; i<=n; i++)
            h[i]=add(h[i-1],1,n,a[i]);
        for (i=1; i<=m; i++) {
            scanf("%d%d%d",&l,&r,&v);
```

```
            ans=ask(h[l-1],h[r],1,n,v);
            ans=f[ans];
            printf("%d\n",ans);
        }
    }
    return 0;
}
```

3 hdu2890(suffixArray)

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<algorithm>
#include<vector>
#include<map>
#include<cstdlib>
#include<ctime>
#include<cstdlib>
#define pb push_back
#define mp make_pair
#define st first
#define nd second
using namespace std;
typedef vector<int>::iterator vit;
typedef map<int,int>::iterator mit;
const int inf=~0u>>2;
const int N=200000;
int n,m;
time_t temp;
struct SA{
    int a[N+1];
    int c[N+1];
    int len;
    int h[N+1];
    pair<int,int> b[N+1];
    int hs[N+1];
    int rk[N+1];
    int tab[N+1];
    int vw[N+1];
    void sa() {
        int i,j,k,x,y;
        memcpy(c,a,(len+1)*sizeof(int));
        b[0]=mp(inf,inf);
        for (i=1; i<=len; i++)
            b[i]=mp(a[i],i);
        sort(b+1,b+len+1);
        for (i=1,j=0; i<=len; i++) {
            if (b[i].st!=b[i-1].st) j++;
            a[b[i].nd]=j;
        }
        memset(hs,0,(len+1)*sizeof(int));
        for (i=1; i<=len; i++)
            hs[a[i]]++;
        for (i=1; i<=len; i++)
            hs[i]+=hs[i-1];
        for (i=len; i; i--)
            tab[hs[a[i]]--]=i;
        for (i=1; i<=len; i<<=1) {
            for (j=len-i+1,k=0; j<=len; j++)
                rk[++k]=j;
            for (j=1; j<=len; j++)
                if (tab[j]>i) rk[++k]=tab[j]-i;
            memset(hs,0,(len+1)*sizeof(int));
            for (j=1; j<=len; j++)
                hs[a[rk[j]]]++;
            for (j=1; j<=len; j++)
                hs[j]+=hs[j-1];
            for (j=len; j; j--)
                tab[hs[a[rk[j]]]--]=rk[j];
            for (j=1,k=0; j<=len; j++) {
                x=tab[j];
                y=tab[j-1];
                if (a[x]!=a[y]||a[x+i]!=a[y+i]) k++;
                vw[tab[j]]=k;
            }
            for (j=1; j<=len; j++) a[j]=vw[j];
        }
        for (i=1; i<=len; i++) {
            if (a[i]==1) h[i]=0;
            else {
                j=tab[a[i]-1];
                k=max(0,h[i-1]-1);
                while (i+k<=len && j+k<=len && c[i+k]==c[j+k]) k++;
                h[i]=k;
            }
        }
    }
} S;
int a[N+1];
int h[N+1];
int q[N+1],top;
int jud(int mid) {
    int i,j,k,t,ret=0;
    for (i=1; i<n; i++) {
```

```
        j=i;
        top=0;
        q[++top]=a[j];
        while (j<n&&h[a[j+1]]>=mid) {
            j++;
            q[++top]=a[j];
        }
        if (j-i+1>=m) {
            sort(q+1,q+top+1);
            ret=0;
            t=-inf;
            for (i=1; i<=top; i++) {
                if (q[i]-t>=mid) ret++,t=q[i];
                if (ret+top-i<m) break;
            }
            if (ret>=m) return q[1];
        }
        i=j;
        if (n-i<m) break;
    }
    return 0;
}
void binary() {
    int l=1,r=n,mid;
    while (l<r) {
        mid=l+r+1>>1;
        if (jud(mid)) l=mid;
        else r=mid-1;
    }
    int i,j;
    j=jud(l);
    if (j) {
        printf("%d\n",l);
        for (i=j; i<=j+l-1; i++)
            printf("%d\n",S.c[i]);
    }
    else printf("0\n");
}
int main() {
    int __,i,x;
    scanf("%d",&__);
    while (__--) {
        scanf("%d%d",&n,&m);

        memset(S.a,0,(2*n+1)*sizeof(int));
        S.len=n;
        for (i=1; i<=n; i++) {
            scanf("%d",&x);
            S.a[i]=x;
        }
        S.sa();
        memcpy(a,S.tab,(n+1)*sizeof(int));
        memcpy(h,S.h,(n+1)*sizeof(int));
        binary();
        if (__) printf("\n");
    }
    return 0;
}
```

4 hdu3691(dinic)

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<queue>
using namespace std;
const int inf=~0u>>1;
int n,m;
int S,T;
int ans;
struct node{
    int to,f,v,next;
} w[100010];
int h[301],cnt=1;
void add(int x,int y,int z) {
    w[++cnt]=(node){y,z,z,h[x]};
    h[x]=cnt;
    w[++cnt]=(node){x,z,z,h[y]};
    h[y]=cnt;
}
queue<int> q;
int d[301];
bool bfs(int S) {
    int i,j,v;
    memset(d,0,sizeof d);
    d[S]=1;
    q.push(S);
    while (!q.empty()) {
        i=q.front();
        q.pop();
        for (v=h[i]; v; v=w[v].next) {
            j=w[v].to;
            if (w[v].f&&!d[j]) {
                d[j]=d[i]+1;
                q.push(j);
```

```
            }
        }
    }
    return d[T];
}
int dfs(int u,int f) {
    int j,v,t,ret=0;
    if (u==T) return f;
    for (v=h[u]; v; v=w[v].next) {
        j=w[v].to;
        if (w[v].f&&d[j]==d[u]+1) {
            t=dfs(j,min(f,w[v].f));
            w[v].f-=t,w[v^1].f+=t;
            f-=t,ret+=t;
            if (!f) return ret;
        }
    }
    return ret;
}
int cal() {
    int i,j,v,ret=0;
    for (v=h[T]; v; v=w[v].next) {
        j=w[v].to;
        ret+=w[v].f-w[v].v;
    }
    return ret;
}
int main() {
    int i,p,x,y,z;
    while (scanf("%d%d%d",&n,&m,&S)!=EOF) {
        if (n==0&&m==0&&S==0) break;
        memset(h,0,sizeof h);
        cnt=1;
        for (i=1; i<=m; i++) {
            scanf("%d%d%d",&x,&y,&z);
            add(x,y,z);
        }
        ans=inf;
        p=S;
        for (i=1; i<=n; i++)
            if (i!=S) {
                T=i;
                while (bfs(p)) dfs(p,inf);
                while (bfs(S)) dfs(S,inf);
                ans=min(ans,cal());
                p=i;
            }
        printf("%d\n",ans);
    }
    return 0;
}
```

5 hdu4992

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
typedef long long ll;
int maxx=1000000;
int n,m;
int p[100001];
bool vst[1000001];
int fst[1000001];
int e[1000001];
void Eular() {
    int i,j,k;
    for (i=2; i<=maxx; i++) {
        if (!vst[i]) {
            p[++p[0]] = i;
            fst[i]=p[0];
            e[i] = i-1;
        }
        for (j=1; j<=p[0] && i*p[j]<=maxx; j++) {
            k=i*p[j];
            vst[k]=1;
            fst[k]=j;
            if (i%p[j]==0) {
                e[k] = e[i] * p[j];
                break;
            }
            else {
                e[k] = e[i] * (p[j] - 1);
            }
        }
    }
    //for (i=1; i<=100; i++) cout<<p[i]<<" ";
    //cout<<endl;
}
int b[101];
void devide(int u) {
    int i,j;
    b[0]=0;
    while (u>1) {
        i=fst[u];
```

```
        j=p[i];
        b[++b[0]]=j;
        while (u%j==0) u/=j;
    }
    //for (i=1; i<=b[0]; i++) cout<<b[i]<<" ";
    //cout<<endl;
}
int pow(int a,int b) {
    int ret=1;
    while (b) {
        if (b&1) ret=(ll)ret*a%n;
        a=(ll)a*a%n;
        b>>=1;
    }
    return ret;
}
bool jud(int u) {
    int i;
    if (pow(u,m)==0) return 0;
    for (i=1; i<=b[0]; i++)
        if (pow(u,m/b[i])==1) return 0;
    return 1;
}
int main() {
    int i;
    Eular();
    while (scanf("%d",&n)!=EOF) {
        m = e[n];
        //cout<<m<<endl;
        //continue;
        devide(m);
        for (i=2; i<n; i++)
            if (jud(i)) printf("%d ",i);
        printf("%d\n");
    }
    return 0;
}
```

6 hdu5081(cdq)

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<algorithm>
#include<cmath>
#include<queue>
#include<map>
#define pb push_back
#define mp make_pair
#define st first
#define nd second
using namespace std;
typedef long long ll;
typedef vector<int>::iterator vit;
typedef vector<pair<int,int> >::iterator pit;
int n,m;
struct Trie{
    int ch[26];
    int fail;
    void init() {
        memset(ch,0,sizeof ch);
        fail=0;
    }
} T[100001];
vector<int> adj[100001];
queue<int> q;
void ac() {
    int i,j,k,fail;
    while (!q.empty()) q.pop();
    T[1].fail=0;
    q.push(1);
    while (!q.empty()) {
        i=q.front();
        q.pop();
        fail=T[i].fail;
        for(j=0; j<26; j++) {
            if (T[i].ch[j]) {
                k=T[i].ch[j];
                T[k].fail=T[fail].ch[j];
                if (!T[k].fail) T[k].fail=1;
                q.push(k);
            }
            else {
                T[i].ch[j]=T[fail].ch[j];
            }
        }
    }
}
int tot;
int f[100001];
int fr[100001];
vector<int> g[100001];
int d[100001];
int fa[100001][21];
void dfs(int u,int v) {
```

```
    int j;
    f[u]=v;
    for (int i=0; i<16; i++)
        if (fa[v][i])
            fa[v][i+1]=fa[fa[v][i]][i];
        else break;
    for (vit it=adj[u].begin(); it!=adj[u].end(); it++) {
        j=*it;
        tot++;
        g[v].pb(tot);
        d[tot]=d[v]+1;
        fa[tot][0]=v;
        dfs(j,tot);
    }
    fr[v]=tot;
}
int lca(int u,int v) {
    int i;
    if (u==v) return u;
    if (d[u]>d[v]) swap(u,v);
    for (i=16; i>=0; i--)
        if (d[fa[v][i]]>=d[u]) v=fa[v][i];
    if (u==v) return u;
    for (i=16; i>=0; i--)
        if (fa[u][i]!=fa[v][i]) u=fa[u][i],v=fa[v][i];
    return fa[u][0];
}
int a[100001];
vector<int> b[100001];
int c[100001];
void ready(int u,int l) {
    int i,j,k;
    sort(c+1,c+l+1);
    if (a[u]==1) {
        j=0;
        for (i=1; i<=l; i++)
        if (c[i]>fr[j]) {
            b[u].pb(c[i]);
            j=c[i];
        }
    }
    else {
        j=c[1];
        b[u].pb(j);
        for (i=2; i<=l; i++) {
            k=lca(j,c[i]);
            b[u].pb(k);
            b[u].pb(c[i]);
            j=c[i];
        }
    }
}
vector<pair<int,int> > vs[400001];
vector<int> vt[400001];
int qs[200001];
void build(int u,int l,int r) {
    int mid;
    if (l==r) {
        vs[u].clear();
        vt[u].clear();
        if (a[l]==1) {
            for (vit it=b[l].begin(); it!=b[l].end(); it++) {
                vs[u].pb(mp(*it,1));
            }
        }
        else {
            int i,j;
            j=0;
            for (vit it=b[l].begin(); it!=b[l].end(); it++) {
                qs[++j]=*it;
            }
            sort(qs+1,qs+j+1);
            for (i=1; i<=j; i++)
                if (qs[i]!=qs[i-1]) vt[u].pb(qs[i]);
        }
        return;
    }
    mid=l+r>>1;
    build(u<<1,l,mid);
    build(u<<1|1,mid+1,r);
    int i,j,l1,l2,lt,rt;

    lt=u<<1;
    rt=u<<1|1;

    vs[u].clear();
    l1=vs[lt].size();
    l2=vs[rt].size();
    i=0,j=0;
    while (i<l1&&j<l2) {
        int t;
        t=vs[lt][i].st-vs[rt][j].st;
        if (t<0) vs[u].pb(vs[lt][i++]);
        if (t>0) vs[u].pb(vs[rt][j++]);
        if (t==0) {
            vs[u].pb(mp(vs[lt][i].st,vs[lt][i].nd+vs[rt][j].nd));
```

```
            i++;
            j++;
        }
    }
    while (i<l1) vs[u].pb(vs[lt][i++]);
    while (j<l2) vs[u].pb(vs[rt][j++]);

    vt[u].clear();
    l1=vt[lt].size();
    l2=vt[rt].size();
    i=0,j=0;
    while (i<l1&&j<l2) {
        int t;
        t=vt[lt][i]-vt[rt][j];
        if (t<0) vt[u].pb(vt[lt][i++]);
        if (t>0) vt[u].pb(vt[rt][j++]);
        if (t==0) {
            vt[u].pb(vt[lt][i]);
            i++;
            j++;
        }
    }
    while (i<l1) vt[u].pb(vt[lt][i++]);
    while (j<l2) vt[u].pb(vt[rt][j++]);

}
struct sta{
    int u,v;
    ll z;
    sta() {}
    sta(int u,int v,ll z):u(u),v(v),z(z) {}
} stk[100001];
int top;
ll sum[100001];
ll ans[100001];
pair<int,int> s[100001];
void cdq(int u,int l,int r) {
    int mid;
    if (l==r) return;
    mid=l+r>>1;
    cdq(u<<1,l,mid);

    if (!vs[u<<1].empty()&&!vt[u<<1|1].empty()) {

        int i,j,k,l1,l2,lt,rt;
        lt=u<<1;
        rt=u<<1|1;
        l1=vs[lt].size();
        l2=vt[rt].size();
        i=0,j=0,k=0;
        while (i<l1&&j<l2) {
            int t;
            t=vs[lt][i].st-vt[rt][j];
            if (t<0) s[++k]=vs[lt][i++];
            if (t>0) s[++k]=mp(vt[rt][j++],0);
            if (t==0) {
                s[++k]=vs[lt][i];
                i++;
                j++;
            }
        }
        while (i<l1) s[++k]=vs[lt][i++];
        while (j<l2) s[++k]=mp(vt[rt][j++],0);

        top=0;
        for (i=1; i<=k; i++) {
            while (top&&s[i].st>fr[stk[top].u]) top--;
            ll t;
            t=(ll)(d[s[i].st]-d[stk[top].u])*stk[top].v+s[i].nd+stk[top].z;
            top++;
            stk[top]=sta(s[i].st,stk[top-1].v+s[i].nd,t);
            sum[s[i].st]=t;
        }

        for (i=mid+1; i<=r; i++)
        if (a[i]==2) {
            int k;
            k=0;
            for (vit it=b[i].begin(); it!=b[i].end(); it++) {
                k^=1;
                if (k) ans[i]+=sum[*it];
                else ans[i]-=sum[*it];
            }
        }

    }

    cdq(u<<1|1,mid+1,r);
}
int main() {
    int __,i,j,k,x;
    char ch;
    scanf("%d",&__);
    while (__--) {
        scanf("%d",&n);
        for (i=1; i<=n; i++)
```

```
            T[i].init();
        for (i=2; i<=n; i++) {
            scanf("%d %c",&x,&ch);
            T[x].ch[ch-'a']=i;
        }
        ac();
        for (i=1; i<=n; i++)
            adj[i].clear();
        for (i=2; i<=n; i++) {
            adj[T[i].fail].pb(i);
        }
        for (i=1; i<=n; i++) g[i].clear();
        tot=1;
        d[1]=1;
        memset(fa,0,sizeof fa);
        dfs(1,1);

        scanf("%d",&m);
        for (i=1; i<=m; i++) b[i].clear();
        memset(ans,0,sizeof ans);
        for (i=1; i<=m; i++) {
            scanf("%d",&a[i]);
            scanf("%d",&k);
            for (j=1; j<=k; j++) {
                scanf("%d",&x);
                c[j]=f[x];
            }
            ready(i,k);
        }
        build(1,1,m);
        cdq(1,1,m);
        for (i=1; i<=m; i++)
            if (a[i]==2)
                printf("%I64d\n",ans[i]);
    }
    return 0;
}
```

7 hdu5127(px+qy)

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<algorithm>
#include<map>
#include<vector>
#define mp make_pair
#define pb push_back
#define st first
#define nd second
using namespace std;
typedef map<pair<int,int>,int>::iterator mit;
typedef long long ll;
typedef pair<int,int> pa;
const ll inf=~0ull>>1;
int n;
int tot;
ll ans[60001];
pair<int,int> a[60001];
map<pair<int,int>,int> b;
pair<pair<int,int>,pair<int,int> > q[60001];
int top;
vector<pair<int,int> > h[200001];
void build(int u,int l,int r) {
    int mid;
    mid=l+r>>1;
    h[u].clear();
    if (l==r) return;
    build(u<<1,l,mid);
    build(u<<1|1,mid+1,r);
}
void add(int u,int l,int r,int lt,int rt,int x,int y) {
    int mid;
    if (l>rt||r<lt) return;
    if (l>=lt&&r<=rt) {
        h[u].pb(mp(x,y));
        return;
    }
    mid=l+r>>1;
    add(u<<1,l,mid,lt,rt,x,y);
    add(u<<1|1,mid+1,r,lt,rt,x,y);
}
bool cmp(const pa &a,const pa &b) {
    return a.nd<b.nd||a.nd==b.nd&&a.st<b.st;
}
pair<int,int> q1[60001];
pair<int,int> q2[60001];
int m1,m2;
pa operator+(const pa &a,const pa &b) {
    return mp(a.st+b.st,a.nd+b.nd);
}
pa operator-(const pa &a,const pa &b) {
    return mp(a.st-b.st,a.nd-b.nd);
}
ll operator*(const pa &a,const pa &b) {
```

```
    return (ll)a.st*b.st+(ll)a.nd*b.nd;
}
ll operator/(const pa &a,const pa &b) {
    return (ll)a.st*b.nd-(ll)a.nd*b.st;
}
ll area(pa p0,pa p1,pa p2) {
    return (p1-p0)/(p2-p0);
}
void scan(int u) {
    int i,l;
    m1=0;
    m2=0;
    l=h[u].size();
    for (i=0; i<l; i++) {
        while (m1>1&&area(q1[m1-1],q1[m1],h[u][i])<=0) m1--;
        q1[++m1]=h[u][i];
    }
    for (i=l-1; i>=0; i--) {
        while (m2>1&&area(q2[m2-1],q2[m2],h[u][i])<=0) m2--;
        q2[++m2]=h[u][i];
    }
}
ll binary(int x,int y,int px,int py) {
    int l,r,mid;
    ll ret;
    l=1;
    r=m1;
    while (l<r) {
        mid=l+r>>1;
        if ((q1[mid+1]-q1[mid])/mp(x,y)>0) l=mid+1;
        else r=mid;
    }
    ret=mp(px,py)*q1[l];
    x=-x;
    y=-y;
    l=1;
    r=m2;
    while (l<r) {
        mid=l+r>>1;
        if ((q2[mid+1]-q2[mid])/mp(x,y)>0) l=mid+1;
        else r=mid;
    }
    ret=max(ret,mp(px,py)*q2[l]);
    return ret;
}
void cal(int u,int l,int r) {
    int mid,i,x,y;
    sort(h[u].begin(),h[u].end(),cmp);
    if (!h[u].empty()) {
        scan(u);
        for (i=l; i<=r; i++) {
            x=-a[i].nd,y=a[i].st;
            if (y<0) x=-x,y=-y;
            ans[i]=max(ans[i],binary(x,y,a[i].st,a[i].nd));
        }
    }
    if (l==r) return;
    mid=l+r>>1;
    cal(u<<1,l,mid);
    cal(u<<1|1,mid+1,r);
}
int main() {
    int i,l,r,t,x,y;
    while (scanf("%d",&n),n) {
        tot=0;
        top=0;
        b.clear();
        for (i=1; i<=n; i++) {
            scanf("%d%d%d",&t,&x,&y);
            if (!t) a[++tot]=mp(x,y);
            else if (t==1) {
                b[mp(x,y)]=tot+1;
            }
            else {
                l=b[mp(x,y)];
                r=tot;
                if (l<=r) q[++top]=mp(mp(x,y),mp(l,r));
                b.erase(mp(x,y));
            }
        }
        for (mit it=b.begin(); it!=b.end(); it++) {
            x=it->st.st;
            y=it->st.nd;
            l=it->nd;
            r=tot;
            if (l<=r) q[++top]=mp(mp(x,y),mp(l,r));
        }
        if (!tot) continue;
        for (i=1; i<=tot; i++) ans[i]=-inf;
        build(1,1,tot);
        for (i=1; i<=top; i++) {
            x=q[i].st.st;
            y=q[i].st.nd;
            l=q[i].nd.st;
            r=q[i].nd.nd;
            add(1,1,tot,l,r,x,y);
```

```
        }
        cal(1,1,tot);
        for (i=1; i<=tot; i++)
            printf("%I64d\n",ans[i]);
    }
    return 0;
}
```

8 hdu5130(圆与多边形交)

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
using namespace std;
const double PI=acos(-1.0);
const double eps=1e-8;
int n;
double k;
double R;
double ans;
int dcmp(double x) {
    if (x<-eps) return -1;
    if (x>eps) return 1;
    return 0;
}
struct point{
    double x,y;
    point() {}
    point(double x,double y):x(x),y(y) {}
    point operator+(const point &a) const{
        return point(x+a.x,y+a.y);
    }
    point operator-(const point &a) const{
        return point(x-a.x,y-a.y);
    }
    double operator*(const point &a) const{
        return x*a.x+y*a.y;
    }
    point operator*(const double &a) const{
        return point(x*a,y*a);
    }
    double operator/(const point &a) const{
        return x*a.y-y*a.x;
    }
    point operator/(const double &a) const{
        return point(x/a,y/a);
    }
    bool operator==(const point &a) const{
        return dcmp(x-a.x)==0&&dcmp(y-a.y)==0;
    }
    double abs() {
        return sqrt(x*x+y*y);
    }
} p[1001];
point O=point(0,0);
point crosspt(point &a,point &b,point &c,point &d) {
    double s1,s2;
    s1=(c-a)/(b-a);
    s2=(b-a)/(d-a);
    return (c*s2+d*s1)/(s1+s2);
}
double sector_area(point a,point b,double r) {
    double theta;
    theta=atan2(a.y,a.x)-atan2(b.y,b.x);
    while (theta <= 0) theta+=2*PI;
    while (theta > 2*PI) theta-=2*PI;
    theta=min(theta,2*PI-theta);
    return r*r*theta/2;
}
void circle_cross_line(point o,double r,point a,point b,point ret[2],int &num) {
    double ax=a.x-o.x;
    double ay=a.y-o.y;
    double bx=b.x-o.x;
    double by=b.y-o.y;
    double dx=bx-ax;
    double dy=by-ay;
    double A=dx*dx+dy*dy;
    double B=2*(ax*dx+ay*dy);
    double C=ax*ax+ay*ay-r*r;
    double delta=B*B-4*A*C;
    num=0;
    if (dcmp(delta)>=0) {
        double t1=(-B-sqrt(delta))/(2*A);
        double t2=(-B+sqrt(delta))/(2*A);
        if (dcmp(t1-1)<=0&&dcmp(t1)>=0)
            ret[num++]=point(a.x+t1*dx,a.y+t1*dy);
        if (dcmp(t2-1)<=0&&dcmp(t2)>=0)
            ret[num++]=point(a.x+t2*dx,a.y+t2*dy);
    }
}
double calc(point &a,point &b) {
    bool ina,inb;
    point ret[2];
    int num;
```

```
    if (0==a||0==b) return 0;
    if (a==b) return 0;
    ina=dcmp((a).abs()-R)<=0;
    inb=dcmp((b).abs()-R)<=0;
    if (ina) {
        if (inb) {
            return abs(a/b)/2.0;
        }
        else {
            circle_cross_line(0,R,a,b,ret,num);
            return abs(a/ret[num-1])/2.0+sector_area(ret[num-1],b,R);
        }
    }
    else{
        if (inb) {
            circle_cross_line(0,R,b,a,ret,num);
            return abs(b/ret[num-1])/2.0+sector_area(ret[num-1],a,R);
        }
        else {
            circle_cross_line(0,R,a,b,ret,num);
            if (num==2) {
                return sector_area(a,ret[0],R)+abs(ret[0]/ret[1])/2.0+sector_area(
                  ret[1],b,R);
            }
            else {
                return sector_area(a,b,R);
            }
        }
    }
}
int main() {
    int i,x,y,T=0;
    point u,v;
    point p1,p2;
    point mid;
    while (scanf("%d%lf",&n,&k)!=EOF) {
        for (i=1; i<=n; i++) {
            scanf("%d%d",&x,&y);
            p[i]=point(x,y);
        }
        scanf("%d%d",&x,&y);
        u=point(x,y);
        scanf("%d%d",&x,&y);
        v=point(x,y);
        p1=v+(u-v)*k/(1+k);
        p2=v+(u-v)*k/(k-1);
        mid=(p1+p2)/2;
        R=(p1-p2).abs()/2;
        for (i=1; i<=n; i++) {
            p[i]=p[i]-mid;
        }
        p[n+1]=p[1];
        ans=0;
        for (i=1; i<=n; i++) {
            int sig;
            sig=dcmp(p[i]/p[i+1]);
            double temp;
            temp=calc(p[i],p[i+1]);
            if (sig>=0) ans+=temp;
            else ans-=temp;
        }
        ans=abs(ans);
        printf("Case %d: %.12f\n",++T,ans);
    }
    return 0;
}
```

9 hdu5384(ac)

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<vector>
#include<queue>
using namespace std;
int n,m;
char ch[20001];
string a[100001];
struct node{
    int ch[26];
    int fa;
    int v;
    int fail;
    void init() {
        fa=0;
        v=0;
        fail=0;
        memset(ch,0,sizeof ch);
    }
} T[200001];
int tot;
void add(int id) {
    int i,j,l,u=0,y;
    l=strlen(ch);
    for (i=0; i<l; i++) {
```

```
        y=ch[i]-'a';
        if (!T[u].ch[y]) {
            T[u].ch[y]=++tot;
            T[tot].init();
            T[tot].fa=u;
        }
        u=T[u].ch[y];
    }
    T[u].v++;
}
queue<int> q;
void ac() {
    int i,j,k,v,fail;
    while (!q.empty()) q.pop();
    T[0].fail=-1;
    q.push(0);
    while (!q.empty()) {
        i=q.front();
        q.pop();
        fail=T[i].fail;
        if (fail!=-1) T[i].v+=T[fail].v;
        for (j=0; j<26; j++)
            if (T[i].ch[j]) {
                v=T[i].ch[j];
                if (fail==-1) T[v].fail=0;
                else T[v].fail=T[fail].ch[j];
                q.push(v);
            }
            else if (fail!=-1) {
                T[i].ch[j]=T[fail].ch[j];
            }
    }
}
void cal(int id) {
    int i,j,k,l,u=0,v,ret=0;
    l=a[id].length();
    for (i=0; i<l; i++) {
        j=a[id][i]-'a';
        u=T[u].ch[j];
        v=T[u].v;
        ret+=v;
    }
    printf("%d\n",ret);
}
void print() {
    int i,j,l,v;
    for (i=0; i<=tot; i++) {
        cout<<i<<":"<<endl;
        for (j=0; j<26; j++)
            if (T[i].ch[j]) cout<<char(j+'a')<<":"<<T[i].ch[j]<<"  ";
        cout<<endl;
    }
}
int main() {
    int __,i;
    scanf("%d",&__);
    while (__--) {
        scanf("%d%d",&n,&m);
        for (i=1; i<=n; i++) {
            scanf("%s",ch);
            a[i]=ch;
        }
        tot=0;
        T[0].init();
        for (i=1; i<=m; i++) {
            scanf("%s",ch);
            add(i);
        }
        ac();
        for (i=1; i<=n; i++) {
            cal(i);
        }
    }
    return 0;
}
```

10 poj2417(bsgs)

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<set>
#include<vector>
using namespace std;
typedef long long ll;
const int mod=100007;
int P,B,N;
int m;
int pow(int a,int b) {
  int ret=1;
  while (b) {
    if (b&1) ret=(ll)ret*a%P;
    a=(ll)a*a%P;
    b>>=1;
```

```
  }
  return ret;
}
set<int> s;
vector<pair<int,int> > hs[100007];
set<int>::iterator it;
void add(int u,int id) {
  int i,j,l;
  j=u%mod;
  l=hs[j].size();
  for (i=0; i<l; i++)
    if (hs[j][i].first==u) {
      hs[j][i].second=id;
      return;
    }
  hs[j].push_back(make_pair(u,id));
  s.insert(j);
}
int ask(int u) {
  int i,j,l;
  j=u%mod;
  l=hs[j].size();
  for (i=0; i<l; i++)
    if (hs[j][i].first==u) {
      return hs[j][i].second;
    }
  return -1;
}
void BSGS() {
  int i,j,k,t;
  m=ceil(sqrt(P));
  for (i=0,j=1; i<m; i++,j=(ll)j*B%P) {
    if (j==N) {
      printf("%d\n",i);
      return;
    }
    add((ll)j*N%P,i);
  }
  for (i=1,k=j; i<=m; i++,k=(ll)k*j%P) {
    t=ask(k);
    if (t!=-1) {
      t=i*m-t;
      printf("%d\n",t);
      return;
    }
  }
  printf("no solution\n");
}
```

```
void init() {
  for (it=s.begin(); it!=s.end(); it++) {
    hs[*it].clear();
  }
  s.clear();
}
int main() {
  s.clear();
  while (scanf("%d%d%d",&P,&B,&N)!=EOF) {
    init();
    BSGS();
  }
  return 0;
}
```

11 tarjan

```
struct node{
    int to,next;
    bool flag;
} w[300001];
int h[100001],cnt=1;
void add(int x,int y) {
    w[++cnt]=(node){y,h[x],0};
    h[x]=cnt;
    w[++cnt]=(node){x,h[y],0};
    h[y]=cnt;
}
int dfn[100001];
int low[100001];
int tot;
void tarjan(int u,int fa) {
    int j,v;
    dfn[u]=low[u]=++tot;
    for (v=h[u]; v; v=w[v].next) {
        j=w[v].to;;
        if (j==fa) continue;
        if (!dfn[j]) {
            tarjan(j,u);
            if (low[j]>dfn[u]) w[v].flag=1,w[v^1].flag=1;
            low[u]=min(low[u],low[j]);
        }
        else low[u]=min(low[u],dfn[j]);
    }
}
```