ACM 模板

from XDU_升华

目录

1、AC 自动机

```cpp
#include <cstdio>
#include <algorithm>
#include <iostream>
#include <cstring>
#include <queue>
using namespace std;
struct Trie
{
    int next[500010][26],fail[500010],end[500010];
    int root,L;
    int newnode()
    {
        for(int i = 0;i < 26;i++)
            next[L][i] = -1;
        end[L++] = 0;
        return L-1;
    }
    void init()
    {
        L = 0;
        root = newnode();
    }
    void insert(char buf[])
    {
        int len = strlen(buf);
        int now = root;
        for(int i = 0;i < len;i++)
        {
            if(next[now][buf[i]-'a'] == -1)
                next[now][buf[i]-'a'] = newnode();
            now = next[now][buf[i]-'a'];
        }
        end[now]++;
    }
    void build()
    {
        queue<int>Q;
        fail[root] = root;
        for(int i = 0;i < 26;i++)
            if(next[root][i] == -1)
                next[root][i] = root;
            else
            {
                fail[next[root][i]] = root;
```

```
45                 Q.push(next[root][i]);
46             }
47         while( !Q.empty() )
48         {
49             int now = Q.front();
50             Q.pop();
51             for(int i = 0;i < 26;i++)
52                 if(next[now][i] == -1)
53                     next[now][i] = next[fail[now]][i];
54                 else
55                 {
56                     fail[next[now][i]]=next[fail[now]][i];
57                     Q.push(next[now][i]);
58                 }
59         }
60     }
61     int query(char buf[])
62     {
63         int len = strlen(buf);
64         int now = root;
65         int res = 0;
66         for(int i = 0;i < len;i++)
67         {
68             now = next[now][buf[i]-'a'];
69             int temp = now;
70             while( temp != root )
71             {
72                 res += end[temp];
73                 end[temp] = 0;
74                 temp = fail[temp];
75             }
76         }
77         return res;
78     }
79     void debug()
80     {
81         for(int i = 0;i < L;i++)
82         {
83             printf("id = %3d,fail = %3d,end = %3d,chi =
84 [",i,fail[i],end[i]);
85             for(int j = 0;j < 26;j++)
86                 printf("%2d",next[i][j]);
87             printf("]\n");
88         }
89     }
90 }ac;
91 char buf[1000010];
92 int main()
93 {
94     int T;
95     int n;
96     scanf("%d",&T);
97     while( T-- )
98     {
99         scanf("%d",&n);
100        ac.init();
101        for(int i = 0;i < n;i++)
102        {
```

```
            scanf("%s",buf);
            ac.insert(buf);
        }
        ac.build();
        scanf("%s",buf);
        printf("%d\n",ac.query(buf));
    }
    return 0;
}
```

2、KMP

```
#include<cstdio>
#include<cstring>
# define M 110000
int i,j,ans,m,n,next[M];
char s[M],t[M];
int main()
{
    ans=0;
    scanf("%s",s+1);
    scanf("%s",t+1);
    n=strlen(s+1);m=strlen(t+1);
    j=0;
    for (i=2;i<=n;i++)
    {
        while (j>0 && s[j+1]!=s[i])
            j=next[j];
        if (s[j+1]==s[i])  j++;
        next[i]=j;
    }
    j=0;
    for (i=1;i<=m;i++)
    {
        while (j>0 && s[j+1]!=t[i])
            j=next[j];
        if (s[j+1]==t[i]) j++;
        if (j==n)
        {
            ans++;j=next[j];
        }
        printf("%d\n",ans);
    }
    //printf("%d\n",ans);
    return 0;
}
```

3、Manachar

```
#include<cstdio>
#define M 110010
char b[M],a[M <<1];
```

```
int p[M <<1];
int i,n,id,maxl,maxid;
int min(int a,int b)
{
    return a<b?a:b;
}
int main()
{
    while (scanf("%s",&b[1])!=EOF)
    {
        maxl=maxid=0;
        for (i=1;b[i]!='\0';i++)
        {
            a[(i<<1)]=b[i];
            a[(i<<1)+1]='#';
            printf("%d\n",i);
        }
        i--;
        a[0]='?';a[1]='#';
        n=(i<<1)+2;a[n+1]=0;
        //a[0],a[n+1]为不同的原串中没有的字符
        for (i=1;i<n;i++)
        {
            if (maxid>i)
                p[i]=min(p[2*id-i],maxid-i);
            else
                p[i]=1;
            while (a[i+p[i]]==a[i-p[i]])
                p[i]++;
            if (p[i]+i>maxid)
            {
                maxid=p[i]+i;
                id=i;
            }
            if (p[i]>maxl) maxl=p[i];
        }
        for (i=0;i<=n;i++)printf("%d ",p[i]-1);//p[i]-1 就是以第 i 个字符为中心的回文串长度
    }
return 0;
}
```

4、后缀数组

```
#include<cstdio>
#include<cstring>
#define N 1000001
int wa[N],wb[N],wv[N],ws[N];
int cmp(int *r,int a,int b,int l)
{return r[a]==r[b]&&r[a+l]==r[b+l];}
void da(int *r,int *sa,int n,int m)
{
    int i,j,p,*x=wa,*y=wb,*t;
```

```
    for(i=0;i<m;i++) ws[i]=0;
    for(i=0;i<n;i++) ws[x[i]=r[i]]++;
    for(i=1;i<m;i++) ws[i]+=ws[i-1];
    for(i=n-1;i>=0;i--) sa[--ws[x[i]]]=i;
    for(j=1,p=1;p<n;j*=2,m=p)
    {
      for(p=0,i=n-j;i<n;i++) y[p++]=i;
      for(i=0;i<n;i++) if(sa[i]>=j) y[p++]=sa[i]-j;
      for(i=0;i<n;i++) wv[i]=x[y[i]];
      for(i=0;i<m;i++) ws[i]=0;
      for(i=0;i<n;i++) ws[wv[i]]++;
      for(i=1;i<m;i++) ws[i]+=ws[i-1];
      for(i=n-1;i>=0;i--) sa[--ws[wv[i]]]=y[i];
      for(t=x,x=y,y=t,p=1,x[sa[0]]=0,i=1;i<n;i++)
      x[sa[i]]=cmp(y,sa[i-1],sa[i],j)?p-1:p++;
    }
    return;
}
/*  DC3╤сие
#define N 1000003
#define F(x) ((x)/3+((x)%3==1?0:tb))
#define G(x) ((x)<tb?(x)*3+1:((x)-tb)*3+2)
int wa[N],wb[N],wv[N],ws[N];
int c0(int *r,int a,int b)
{return r[a]==r[b]&&r[a+1]==r[b+1]&&r[a+2]==r[b+2];}
int c12(int k,int *r,int a,int b)
{if(k==2) return r[a]<r[b]||r[a]==r[b]&&c12(1,r,a+1,b+1);
 else return r[a]<r[b]||r[a]==r[b]&&wv[a+1]<wv[b+1];}
void sort(int *r,int *a,int *b,int n,int m)
{
    int i;
    for(i=0;i<n;i++) wv[i]=r[a[i]];
    for(i=0;i<m;i++) ws[i]=0;
    for(i=0;i<n;i++) ws[wv[i]]++;
    for(i=1;i<m;i++) ws[i]+=ws[i-1];
    for(i=n-1;i>=0;i--) b[--ws[wv[i]]]=a[i];
    return;
}
void dc3(int *r,int *sa,int n,int m)
{
    int i,j,*rn=r+n,*san=sa+n,ta=0,tb=(n+1)/3,tbc=0,p;
    r[n]=r[n+1]=0;
    for(i=0;i<n;i++) if(i%3!=0) wa[tbc++]=i;
    sort(r+2,wa,wb,tbc,m);
    sort(r+1,wb,wa,tbc,m);
    sort(r,wa,wb,tbc,m);
    for(p=1,rn[F(wb[0])]=0,i=1;i<tbc;i++)
    rn[F(wb[i])]=c0(r,wb[i-1],wb[i])?p-1:p++;
    if(p<tbc) dc3(rn,san,tbc,p);
    else for(i=0;i<tbc;i++) san[rn[i]]=i;
    for(i=0;i<tbc;i++) if(san[i]<tb) wb[ta++]=san[i]*3;
    if(n%3==1) wb[ta++]=n-1;
    sort(r,wb,wa,ta,m);
    for(i=0;i<tbc;i++) wv[wb[i]=G(san[i])]=i;
    for(i=0,j=0,p=0;i<ta && j<tbc;p++)
    sa[p]=c12(wb[j]%3,r,wa[i],wb[j])?wa[i++]:wb[j++];
    for(;i<ta;p++) sa[p]=wa[i++];
    for(;j<tbc;p++) sa[p]=wb[j++];
```

```
    return;
}
*/
int rank[N],height[N];
void calheight(int *r,int *sa,int n)
{
    int i,j,k=0;
    for(i=1;i<=n;i++) rank[sa[i]]=i;
    for(i=0;i<n;height[rank[i++]]=k)
    for(k?k--:0,j=sa[rank[i]-1];r[i+k]==r[j+k];k++);
    return;
}
int RMQ[N];
int mm[N];
int best[20][N];
void initRMQ(int n)
{
    int i,j,a,b;
    for(mm[0]=-1,i=1;i<=n;i++)
    mm[i]=((i&(i-1))==0)?mm[i-1]+1:mm[i-1];
    for(i=1;i<=n;i++) best[0][i]=i;
    for(i=1;i<=mm[n];i++)
    for(j=1;j<=n+1-(1<<i);j++)
    {
      a=best[i-1][j];
      b=best[i-1][j+(1<<(i-1))];
      if(RMQ[a]<RMQ[b]) best[i][j]=a;
      else best[i][j]=b;
    }
    return;
}
int askRMQ(int a,int b)
{
   int t;
   t=mm[b-a+1];b-=(1<<t)-1;
   a=best[t][a];b=best[t][b];
   return RMQ[a]<RMQ[b]?a:b;
}
int lcp(int a,int b)
{
   int t;
   a=rank[a];b=rank[b];
   if(a>b) {t=a;a=b;b=t;}
   return(height[askRMQ(a+1,b)]);
}
int ans, ,n,i,a[N],sa[N];
char s[N];
void debug()
{
   int i;
   for (i=0;i<n;i++)
      printf("%d ",sa[i]);
   printf("%d\n",sa[n]);
   for (i=0;i<n;i++)
      printf("%d ",rank[i]);
   printf("%d\n",rank[n]);
   for (i=0;i<n;i++)
      printf("%d ",height[i]);
```

```
    printf("%d\n",height[i]);
}

int main()
{
    scanf("%d",& );
    while ( --)
    {
        ans=0;
        scanf("%s",s);
        n=strlen(s);
        for (i=0;i<n;i++)
            a[i]=s[i];
        a[n]=0;
        da(a,sa,n+1,300);
        calheight(a,sa,n);
        debug();
        for (i=1;i<=n;i++)
            ans+=n-sa[i]-height[i];
        printf("%d\n",ans);
    }
    return 0;
}
```

5、最大流

```
#include<cstdio>
#include<cstring>
#define M 210
#define N 210
int len[M <<1],e[M <<1],nex[M <<1],other[M
<<1],head[N],last[N],d[N],num[N];
int m,n,i,ans,tot,ss,tt,ee,u,v,c;
int init()
{
    memset(head,0,sizeof(head));
    memset(num,0,sizeof(num));
    memset(d,0,sizeof(d));
    ans=ee=0;
}
int min(int a,int b)
{
    return a<b?a:b;
}
void add(int u,int v,int c)
{
    other[++ee]=ee+1;
    e[ee]=v;nex[ee]=head[u];head[u]=ee;len[ee]=c;
    other[++ee]=ee-1;
    e[ee]=u;nex[ee]=head[v];head[v]=ee;len[ee]=0;
}
int dfs(int x,int flow)
{
```

```
    int rec,j,p;
    if (x==tt) return flow;
    rec=0;j=last[x];
    while (j!=0)
    {
        if (len[j]>0 && d[x]==d[e[j]]+1)
        {
            last[x]=j;
            p=dfs(e[j],min(len[j],flow-rec));
            len[j]-=p;len[other[j]]+=p;
            rec+=p;
            if (rec==flow) return rec;
        }
    j=nex[j];
    }
    if (d[ss]>tot) return rec;
    if (--num[d[x]]==0) d[ss]=tot;
    last[x]=head[x];
    num[++d[x]]++;
    return rec;
}
int main()
{
    while (scanf("%d%d",&m,&n)!=EOF)
    {
        init();
        for (i=1;i<=m;i++)
        {
            scanf("%d%d%d",&u,&v,&c);
            add(u,v,c);
        }
        tot=num[0]=n;
        ss=1;tt=n;
        for (i=ss;i<=tt;i++)
            last[i]=head[i];
        while (d[ss]<tot)
            ans+=dfs(ss,2147483647);
        printf("%d\n",ans);
    }
    return 0;
}
```

6、最小费用最大流

```
#include<cstdio>
#include<cstring>
#define INF 1000000000
#define N 1010
#define M 30010
#define WW 10000
int _,other[M],e[M],x[M],nex[M],len[M],ww[M];

```

```
int
pre[N],vis[N],head[N],dis[N],a[5000000],ss,tt,n,m,i,j,k,opt,maxflow,mincos
t,ee;
int xx,xq[300][300],gy[300][300],sum,summ,flag;
int min(int a,int b)
{return a<b?a:b;}

bool spfa()
{
    int h,t,i,j;
    h=1;t=1;
    for (i=0;i<=tt;i++) dis[i]=INF;
    a[1]=ss;pre[ss]=-1;
    vis[ss]=1;dis[ss]=0;
    while (h<=t)
    {
        j=head[a[h]];
        while (j!=0)
        {
            if (len[j]>0 && dis[a[h]]+ww[j]<dis[e[j]])
            {
                dis[e[j]]=dis[a[h]]+ww[j];
                pre[e[j]]=j;
                if (vis[e[j]]==0)
                {
                    vis[e[j]]=1;a[++t]=e[j];
                }
            }
            j=nex[j];
        }
        vis[a[h++]]=0;
    }
    if (dis[tt]>=INF-1000) return 0;
    else return 1;
}

void work()
{
    int zh,flow;
    mincost=maxflow=0;
    while (spfa())
    {
        zh=tt;
        flow=INF;
        while (pre[zh]!=-1)
        {
            flow=min(flow,len[pre[zh]]);
            zh=x[pre[zh]];
        }
        maxflow+=flow;
        mincost+=dis[tt]*flow;
        zh=tt;
        while (pre[zh]!=-1)
        {
            len[pre[zh]]-=flow;
            len[other[pre[zh]]]+=flow;
            zh=x[pre[zh]];
        }
```

```
    }
}

void add(int u,int v,int c,int w)
{
    e[++ee]=v;x[ee]=u;nex[ee]=head[u];head[u]=ee;
    len[ee]=c;ww[ee]=w;other[ee]=ee+1;
    e[++ee]=u;x[ee]=v;nex[ee]=head[v];head[v]=ee;
    len[ee]=0;ww[ee]=-w;other[ee]=ee-1;
}

void init()
{
    memset(head,0,sizeof(head));
    memset(nex,0,sizeof(nex));
    sum=ee=0;
}
int main()
{
    scanf("%d",& );
    while ( --)
    {
        init();
        n=3;
        add(1,2,1,-1);
        add(2,3,1,-1);
        add(3,1,1,-1);
        ss=1;tt=n;
        work();
    }
    return 0;
}
```

7、最小树形图-朱刘算法

```
#include <cstdio>
#include <iostream>
#include <cmath>
#include <cstring>
using namespace std;
const int N=101,M=10001,inf=2147483647;
struct edge {int u,v;double w;} e[M];
int i,n,m,id[N],pre[N],v[N];
double x[N],y[N],ans,inw[N];
void zhu liu(int root)
{
   int s,t,idx=1;
   while (idx)
     {
       for (int i=1;i<=n;++i) inw[i]=inf,id[i]=0,v[i]=0;
       for (int i=1;i<=m;++i)
         {
           s=e[i].u;t=e[i].v;
```

```
            if (e[i].w>=inw[t] || s==t) continue;
            pre[t]=s;
            inw[t]=e[i].w;
          }
        //for (i=1;i<=n;i++)printf("%f\n",inw[i]);
        inw[root]=0;pre[root]=root;
        for (int i=1;i<=n;++i)
          {
            if (inw[i]==inf)
              {
                printf("poor snoopy\n");
                return;
              }
            ans+=inw[i];
          }
        idx=0;
        for (int i=1;i<=n;++i)
          if (v[i]==0)
            {
              t=i;
              while (v[t]==0) v[t]=i,t=pre[t];
              if (v[t]!=i || t==root) continue;
              id[t]=++idx;
              for (s=pre[t];s!=t;s=pre[s]) id[s]=idx;
            }
        if (idx==0) continue;
        for (int i=1;i<=n;++i)
          if (id[i]==0) id[i]=++idx;
        for (int i=1;i<=m;++i)
          {
            e[i].w-=inw[e[i].v];
            e[i].u=id[e[i].u];
            e[i].v=id[e[i].v];
          }
        n=idx;
        root=id[root];
      }
    printf("%.2f\n",ans);
}
int main()
{
    while (scanf("%d%d",&n,&m)==2)
    {
        ans=0;
        for (i=1;i<=n;i++)
            scanf("%lf%lf",&x[i],&y[i]);
        for (i=1;i<=m;i++)
        {
            scanf("%d%d",&e[i].u,&e[i].v);
            e[i].w=sqrt((x[e[i].u]-x[e[i].v])*(x[e[i].u]-
x[e[i].v])+(y[e[i].u]-y[e[i].v])*(y[e[i].u]-y[e[i].v]));
        }
    zhu liu(1);
    }
    return 0;
}
```

8、Tarjan 强连通分量

```
#include<cstdio>
#include<cstring>
#define M 100001
#define N 100001
int stap[N],e[M],next[M],dfn[N],head[N],low[N],c[N];
bool instap[N];
int color,time,stop,i,n,m,u,v,ee;
int min(int a,int b)
{
    return a<b?a:b;
}
void tarjan(int u)
{
    int j;
    dfn[u]=low[u]=++time;
    instap[u]=1;
    stap[++stop]=u;
    for (j=head[u];j;j=next[j])
    {
        if (!dfn[e[j]])
        {
            tarjan(e[j]);
            low[u]=min(low[u],low[e[j]]);
        }
        else if (instap[e[j]])
            low[u]=min(low[u],dfn[e[j]]);
    }
    if (dfn[u]==low[u])
    {
        color++;
        while (1)
        {
            j=stap[stop--];
            instap[j]=0;
            c[j]=color;
            if (j==u) break;
        }
    }
}
void add(int u,int v)
{
    e[++ee]=v;next[ee]=head[u];head[u]=ee;
}
void init()
{
    ee=stop=color=time=0;
    memset(head,0,sizeof(head));
    memset(next,0,sizeof(next));
    memset(dfn,0,sizeof(dfn));
    memset(instap,0,sizeof(instap));
}
int main()
{
    while (scanf("%d%d",&n,&m)==2)
```

```
    {
        init();
        for (i=1;i<=m;i++)
        {
            scanf("%d%d",&u,&v);
            add(u,v);
        }
        for (i=1;i<=n;i++)
            if (!dfn[i]) tarjan(i);
        for (i=1;i<=n;i++)
            printf("%d ",c[i]);
    }
return 0;
}
```

9、Tarjan 求割边割点

```
/*if (ll[e[i]]>pre[u]) i 是割边*/
#include<cstdio>
#include<cstring>
#define M 100001
#define N 100001
int e[M],next[M],dfn[N],head[N],low[N];
int ee,du,i,n,m,u,v,time;
bool vis[M];
inline int min(int a,int b)
{
    return a<b?a:b;
}
void tarjan(int u)
{
    int j;
    low[u]=dfn[u]=++time;
    for (j=head[u];j;j=next[j])
        if (!vis[j])
        {
            vis[j]=1;vis[((j+1)^1)-1]=1;
            if (dfn[e[j]]==0)
            {
                tarjan(e[j]);
                low[u]=min(low[u],low[e[j]]);
                if (u==1) du++;
                if (low[e[j]]>=dfn[u])  printf("%d\n",u);
            }
            else low[u]=min(low[u],dfn[e[j]]);
        }
}
void add(int u,int v)
{
    e[++ee]=v;next[ee]=head[u];head[u]=ee;
    e[++ee]=u;next[ee]=head[v];head[v]=ee;
}
void init()
```

```
{
    time=du=ee=0;
    memset(head,0,sizeof(head));
    memset(next,0,sizeof(next));
    memset(dfn,0,sizeof(dfn));
    memset(vis,0,sizeof(vis));
    memset(low,0,sizeof(low));
}
int main()
{
    while (scanf("%d%d",&n,&m)==2)
    {
        init();
        for (i=1;i<=m;i++)
        {
            scanf("%d%d",&u,&v);
            add(u,v);
        }
        tarjan(1);
    }
    return 0;
}
```

10、无源汇的最小割-Stoer-Wagner 算法

```
#include <cstdio>
#include <cstring>
#define INF 1000000000
#define N 510
using namespace std;

int mp[N][N];
int i,n,m,u,v,c;
bool combine[N];
int minC=INF;

void search(int &s,int &t)
{
    bool vis[N];
    int i,j,w[N];
    memset(vis,0,sizeof(vis));
    memset(w,0,sizeof(w));
    int tmpj=1000;
    for(i=1;i<=n;i++)
    {
        int max=-INF;
        for(j=1;j<=n;j++)
        {
            if(!vis[j]&&!combine[j]&&max<w[j])
            {
                max=w[j];
                tmpj=j;
            }
```

```
        }
        if(t==tmpj){minC=w[t];return;}
        vis[tmpj]=1;
        s=t,t=tmpj;
        for(int j=1;j<=n;j++)
        {
            if(!vis[j]&&!combine[j])
                w[j]+=mp[t][j];
        }
    }
    minC=w[t];
}

int mincut()
{
    int ans=INF,i,j,s,t;
    memset(combine,0,sizeof(combine));
    for(i=1;i<=n-1;i++)
    {
        s=t=-1;
        search(s,t);
        combine[t]=true;
        ans=ans>minC?minC:ans;
        for(j=1;j<=n;j++)
        {
            mp[s][j]+=mp[t][j];
            mp[j][s]+=mp[j][t];
        }
    }
    return ans;
}
int main()
{
    while(scanf("%d%d",&n,&m)==2)
    {
        memset(mp,0,sizeof(mp));
        for(i=1;i<=m;i++)
        {
            scanf("%d%d%d",&u,&v,&c);
            u++;v++;
            mp[u][v]+=c;
            mp[v][u]+=c;
        }
        printf("%d\n",mincut());
    }
    return 0;
}
```

11、倍增 LAC

```
#include<cstdio>
#include<cstring>
#include<algorithm>
```

```cpp
#include<cmath>
#include<string>
#include<iostream>
#include<map>
#define MAX(a,b) a>b?a:b
#define N 110000
#define mem(a) memset(a,0,sizeof(a))
using namespace std;
int l[N<<1],sum,j,k,deep,hh,tt,ee,i,n,ans,e[N<<1],head[N],nex[N<<1];
int q,f[N],a[N<<2],u,v,c,dep[N],len[N][20],p[N][20];
void add(int u,int v,int c)
{
    e[++ee]=v;nex[ee]=head[u];head[u]=ee;l[ee]=c;
}
int lca(int a,int b)
{
    int i,j,k;
    if (dep[a]>dep[b])
        swap(a,b);
    k=(int)(log(dep[b])/log(2));
    for (i=k;i>=0;i--)
        if (dep[b]-(1<<i)>=dep[a])
        {
            sum+=len[b][i];
            b=p[b][i];
        }
    if (a==b) return a;
    k=(int)(log(dep[b])/log(2));
    for (i=k;i>=0;i--)
        if (p[a][i]!=p[b][i])
        {
            sum+=len[b][i]+len[a][i];
            a=p[a][i];
            b=p[b][i];
        }
    sum+=len[a][0]+len[b][0];
    return f[a];
}

int main()
{
    while (scanf("%d%d",&n,&q)==2)
    {
        mem(head);mem(dep);mem(p);
        ee=0;
        for (i=1;i<=n-1;i++)
        {
            scanf("%d%d%d",&u,&v,&c);
            add(u,v,c);
            add(v,u,c);
        }
        a[1]=1;hh=tt=1;dep[1]=1;
        while (hh<=tt)
        {
            j=head[a[hh]];
            while (j>0)
            {
                if (dep[e[j]]==0)
```

```
                {
                    dep[e[j]]=dep[a[hh]]+1;
                    deep=MAX(deep,dep[e[j]]);
                    f[e[j]]=a[hh];
                    len[e[j]][0]=l[j];
                    a[++tt]=e[j];
                }
                j=nex[j];
            }
            hh++;
        }
        for (i=1;i<=n;i++)
            p[i][0]=f[i];
        k=(int)(log(deep)/log(2));
        for (j=1;j<=k;j++)
            for (i=1;i<=n;i++)
            {
                p[i][j]=p[p[i][j-1]][j-1];
                len[i][j]=len[i][j-1]+len[p[i][j-1]][j-1];
            }
    }
    return 0;
}
```

12、树链剖分

```
#include<cstdio>
#include<cstring>
#include<algorithm>
#define MAX(a,b) a>b?a:b
#define N 110000
#define mem(a) memset(a,0,sizeof(a))
using namespace std;
int
uu,vv,ma,root,ee,rr,ll, ,i,n,tot,ans,e[N<<1],head[N],len[N<<1],nex[N<<1];
int
top[N],siz[N],s[N],d[N],w[N],f[N],l[N<<2],r[N<<2],a[N<<2],u[N],v[N],c[N];
char opt[100];
void add(int u,int v,int c)
{
    e[++ee]=v;nex[ee]=head[u];head[u]=ee;
}
void build(int s,int ll,int rr)
{
    l[s]=ll;r[s]=rr;
    if (ll==rr)
        a[s]=0;
    else
    {
        int mid=(ll+rr)>>1;
        build(s<<1,ll,mid);
        build((s<<1)+1,mid+1,rr);
    }
```

```
}
void add(int s)
{
    if (l[s]==r[s])
        a[s]=rr;
    else
    {
        if (r[s<<1]>=ll)
            add(s<<1);
        else
            add((s<<1)+1);
        a[s]=MAX(a[s<<1],a[(s<<1)+1]);
    }
}
void sea(int s)
{
    if (l[s]>rr || r[s]<ll)
        return;
    if (l[s]>=ll&&r[s]<=rr)
        ans=MAX(ans,a[s]);
    else
    {
        sea(s<<1);
        sea((s<<1)+1);
    }
}
void dfs1(int u)
{
    int j=head[u];
    siz[u]=1;
    while (j>0)
    {
        if (d[e[j]]==0)
        {
            d[e[j]]=d[u]+1;
            f[e[j]]=u;
            dfs1(e[j]);
            if (siz[e[j]]>siz[s[u]])
                s[u]=e[j];
            siz[u]+=siz[e[j]];
        }
        j=nex[j];
    }
}
void dfs2(int u)
{
    int j=head[u];
    if (s[u]!=0)
    {
        w[s[u]]=++tot;
        top[s[u]]=top[u];
        dfs2(s[u]);
    }
    while (j>0)
    {
        if (e[j]!=s[u]&&e[j]!=f[u])
        {
            w[e[j]]=++tot;
```

```
            top[e[j]]=e[j];
            dfs2(e[j]);
        }
        j=nex[j];
    }
}
void init()
{
    mem(head);
    mem(s);mem(a);mem(d);
    ee=tot=0;
}
int main()
{
    scanf("%d",& );
    while ( --)
    {
        init();
        scanf("%d",&n);
        build(1,1,n-1);
        for (i=1;i<=n-1;i++)
        {
            scanf("%d%d%d",&u[i],&v[i],&c[i]);
            add(u[i],v[i],c[i]);
            add(v[i],u[i],c[i]);
        }
        root=1;d[root]=1;top[root]=root;
        dfs1(root);
        dfs2(root);
        for (i=1;i<=n-1;i++)
        {
            if (d[u[i]]>d[v[i]])
                swap(u[i],v[i]);
            ll=w[v[i]];rr=c[i];
            add(1);
        }
        while (1)
        {
            getchar();
            scanf("%s",opt);
            if (opt[0]=='D')break;
            scanf("%d%d",&uu,&vv);
            if (opt[0]=='Q')
            {
                ma=0;
                while (top[uu]!=top[vv])
                {
                    if (d[top[uu]]<d[top[vv]])
                        swap(uu,vv);
                    ans=0;ll=w[top[uu]];rr=w[uu];sea(1);
                    ma=MAX(ma,ans);
                    uu=f[top[uu]];
                }
                if (uu!=vv)
                {
                    if (d[uu]<d[vv]) swap(uu,vv);
                    ans=0;ll=w[s[vv]];rr=w[uu];sea(1);
                    ma=MAX(ma,ans);
```

```
                }
                printf("%d\n",ma);
            }
            if (opt[0]=='C')
            {
                ll=w[v[uu]];rr=vv;
                add(1);
            }
        }
    }
    return 0;
}
/*
14
1 2
2 5
2 6
6 11
6 12
1 3
3 7
1 4
4 8
4 9
4 10
9 13
13 14
*/
```

13、FFT 快速卷积

```
#include <cstdio>
#include <cstring>
#include <iostream>
#include <algorithm>
#include <cmath>
#define PI acos(-1.0)
#define int long long
#define fi first
#define se second
#define mp make pair
using namespace std;
typedef long long ll;
const int N=60010;
ll a[N*16],b[N*16],n;
pair <ll,ll> e[N*16];
//复数结构体
struct complex
{
    long double r,i;
    complex(long double  r = 0.0,long double  i = 0.0)
    {
        r = _r; i = _i;
```

```
    }
    complex operator +(const complex &b)
    {
        return complex(r+b.r,i+b.i);
    }
    complex operator -(const complex &b)
    {
        return complex(r-b.r,i-b.i);
    }
    complex operator *(const complex &b)
    {
        return complex(r*b.r-i*b.i,r*b.i+i*b.r);
    }
};
void change(complex y[],int len)
{
    int i,j,k;
    for(i = 1, j = len/2;i < len-1; i++)
    {
        if(i < j) swap(y[i],y[j]);
        //交换互为小标反转的元素，i<j 保证交换一次
        //i 做正常的+1，j 左反转类型的+1,始终保持 i 和 j 是反转的
        k = len/2;
        while( j >= k)
        {
            j -= k;
            k /= 2;
        }
        if(j < k) j += k;
    }
}
/*
 * 做 FFT
 * len 必须为 2^k 形式，
 * on==1 时是 DFT，on===-1 时是 IDFT
 */
void fft(complex y[],int len,int on)
{
    change(y,len);
    for(int h = 2; h <= len; h <<= 1)
    {
        complex wn(cos(-on*2*PI/h),sin(-on*2*PI/h));
        for(int j = 0;j < len;j+=h)
        {
            complex w(1,0);
            for(int k = j;k < j+h/2;k++)
            {
                complex u = y[k];
                complex t = w*y[k+h/2];
                y[k] = u+t;
                y[k+h/2] = u-t;
                w = w*wn;
            }
        }

    }
    if(on == -1)
        for(int i = 0;i < len;i++)
```

```
            y[i].r /= len;
    }
}
const int MAXN = N*16;
complex x1[MAXN],x2[MAXN];
int sum[MAXN];
bool cmp(pair <ll,ll> a,pair <ll,ll> b){
    return (a.fi>b.fi);
}
main()
{
    int  ;
    scanf("%lld",& );
    while( --)
    {
        scanf("%lld",&n);
        ll s=0;
        memset(a,0,sizeof(a));
        memset(b,0,sizeof(b));
        for (ll i=0;i<n;i++)
            scanf("%lld",&a[i]),s+=a[i]*a[i];
        for (ll i=0;i<n;i++)
            scanf("%lld",&b[i]),s+=b[i]*b[i];
        for (ll i=0;i<n/2;i++)
            swap(b[i],b[n-i-1]);//因为后面 n 变了
        ll x=1;
        while (x<n*2) x<<=1;
        ll nn=n;
        ll len=n=x;
        for(int i = 0;i < n;i++)
            x1[i] = complex(a[i],0);
        for(int i = 0;i < n;i++)
            x2[i] = complex(b[i],0);
        //求 DFT
        fft(x1,len,1);
        fft(x2,len,1);
        for(int i = 0;i < len;i++)
            x1[i] = x1[i]*x2[i];
        fft(x1,len,-1);
        for(int i = 0;i < len;i++){
            sum[i] = (long long)(x1[i].r+0.5);
        }
        ll ans=0;
        n=nn;
        for (ll i=0;i<n;i++)
            e[i]=mp(sum[i]+sum[i+n],i);
        sort(e,e+n,cmp);
        for (ll i=n;i<n*2;i++)
            a[i]=a[i-n];
        for (ll i=0;i<min(100ll,n);i++){
            ll t=e[i].se+1,p=0;
            for (ll j=0;j<n;j++)
                p+=a[j+t]*b[n-1-j];
            ans=max(ans,p);
        }
        printf("%lld\n",s-ans*2);
    }
    return 0;
}
```

14、Lucas 定理求组合数

```
#include<cstdio>
#define P 10007
long long n,m;
long long pow(long long a,long long b)
{
    long long ans;
    ans=1;
    while (b>0)
    {
        if (b%2)
            ans=(ans*a)%P;
        b=b>>1;
        a=(a*a)%P;
    }
    return ans;
}
long long c(long long n,long long m)
{
    long long cc,i;
    if (n<m) return 0;
    cc=1;
    for (i=n;i>=n-m+1;i--) cc=cc*i%P;
    for (i=2;i<=m;i++)
        cc=cc*pow(i,P-2)%P;
    return cc;
}
long long  lucas(long long n,long long m)
{
    if (m==0) return 1;
    return c(n%P,m%P)*lucas(n/P,m/P)%P;
}
int main()
{
    while(scanf("%lld%lld",&n,&m)==2)
    {
        printf("%lld\n",lucas(n,m));
    }
    return 0;
}
```

15、欧拉筛法

```
#include<cstdio>
#include<cstring>
#define M 10000000
int pp,i,j,n,phi[M],p[M];
```

```cpp
bool b[M];
int main()
{
    freopen("1.txt","w",stdout);
    memset(b,0,sizeof(b));
    scanf("%d",&n);
    phi[1]=1;
    for (i=2;i<=n;i++)
    {
        if (!b[i])
        {
            p[++pp]=i;
            phi[i]=i-1; //phi
        }
        for (j=1;j<=pp;j++)
        {
            if (p[j]*i>n) break;
            b[p[j]*i]=true;
            if (i%p[j]==0)
            {
                phi[p[j]*i]=phi[i]*p[j];  //phi
                break;
            }
            else
                phi[p[j]*i]=phi[i]*(p[j]-1); //phi
        }
    }
    for (i=1;i<=pp;i++) printf("%d,",p[i]);
    printf("%d",pp);
    return 0;
}
```

16、高斯消元

```cpp
#include<cstdio>
#include<cstring>
#include<algorithm>
#include<cmath>
#define MAX(a,b) a>b?a:b
#define N 1100
#define NU 1000000
#define mem(a) memset(a,0,sizeof(a))
using namespace std;
double x[N],ma[N][N],a[N][N],m;
int i,j,k,n;
int main()
{
    scanf("%d",&n);
    for (i=1;i<=n+1;i++)
        for (j=1;j<=n;j++)
            scanf("%lf",&a[i][j]);
    for (i=1;i<=n;i++)
        for (j=1;j<=n;j++)
```

```
        {
            ma[i][j]=(a[i+1][j]-a[i][j])*2;
            ma[i][n+1]+=a[i+1][j]*a[i+1][j]-a[i][j]*a[i][j];
        }

    for (i=1;i<=n-1;i++)
    {
        for (j=i;j<=n;j++)
            if (fabs(ma[j][i])>fabs(ma[i][i]))
                for (k=1;k<=n+1;k++)
                    swap(ma[i][k],ma[j][k]);
        if (ma[i][i]==0)continue;
        for (j=i+1;j<=n;j++)
        {
            m=ma[j][i]/ma[i][i];
            ma[j][i]=0;
            for (k=i+1;k<=n+1;k++)
                ma[j][k]=ma[i][k]*m-ma[j][k];
        }
    }
    x[n]=ma[n][n+1]/ma[n][n];
    for (i=n-1;i>=1;i--)
    {
        m=0;
        for (j=i+1;j<=n;j++)
            m+=ma[i][j]*x[j];
        x[i]=(ma[i][n+1]-m)/ma[i][i];
    }
    for (i=1;i<n;i++)
        printf("%.3f ",x[i]);
    printf("%.3f\n",x[n]);
}
```

17、树状数组

```
int lowbit(int x)
{
    return (x&(-x));
}
int search(int x)
{
    int sea=0;
    while (x>0)
    {
        sea+=c[x];
        x-=lowbit(x);
    }
    return sea;
}
void change(int x,int a)
{
    while (x<=n)
    {
```

```
        c[x]+=a;
        x+=lowbit(x);
    }
}
```

18、二维树状数组

```
#include<cstdio>
#include<cstring>
int c[1100][1100],n,p,i,j, ,x1,x2,y1,y2;
char opt[10];
int lowbit(int x)
{
    return x&(-x);
}
int search(int x,int y)
{
    int i=x,j=y,ans=0;
    while (i>0)
    {
        j=y;
        while (j>0)
        {
            ans+=c[i][j];
            j-=lowbit(j);
        }
        i-=lowbit(i);
    }
    return ans;
}
void change(int x,int y,int v)
{
    int i=x;j=y;
    while (i<=n)
    {
        j=y;
        while (j<=n)
        {
            c[i][j]+=v;
            j+=lowbit(j);
        }
        i+=lowbit(i);
    }
}
int main()
{
    scanf("%d",& );
    while ( --)
    {
        memset(c,0,sizeof(c));
        scanf("%d%d",&n,&p);
        for (i=1;i<=p;i++)
        {
```

```
            scanf("%s",opt);
            if (opt[0]=='Q')
            {
                scanf("%d%d",&x1,&y1);
                printf("%d\n",search(x1,y1)%2);
            }
            if (opt[0]=='C')
            {
                scanf("%d%d%d%d",&x1,&y1,&x2,&y2);
                change(x1,y1,1);
                change(x1,y2+1,1);
                change(x2+1,y1,1);
                change(x2+1,y2+1,1);
            }
        }
        if ( >0) puts("");
    }
    return 0;
}
```

19、小根堆

```
void up()
{
    int i=n;
    while (a[i]<a[i>>1] && (i>>1)>0)
    {
        swap(a[i],a[i>>1]);
        i>>=1;
    }
}
void down()
{
    int now=1,i;
    while (now<=n/2)
    {
        i=now*2;
        if (i<n&&a[i]>a[i+1]) i++;
        if (a[now]>a[i])
        {
            swap(a[i],a[now]);
            now=i;
        }
        else
            break;
    }
}
```

20、圆面积交

```
double calc(double x1,double y1,double r1,double x2,double y2,double r2)
{
    double d=sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
    if(r1>r2)
    {
        double temp=r1;
        r1=r2;
        r2=temp;
    }
    if(r1+r2<=d)
        return 0;
    else if(r2-r1>=d)
        return pi*r1*r1;
    else
     {
        double a1=acos((r1*r1+d*d-r2*r2)/(2.0*r1*d));
        double a2=acos((r2*r2+d*d-r1*r1)/(2.0*r2*d));
        return (a1*r1*r1+a2*r2*r2-r1*d*sin(a1));
    }
}
```

21、Contor 展开

```
#include <cstdio>
#include <cstring>
using namespace std;
const int fac[8] = {1,1,2,6,24,120,720,5040};
struct node
{
    int a[8];
    int n;
}u,v;
int contor(node &t)
{
    int tmp,num=0;
    for (int i = 0; i < 8; i++)
    {
        tmp = 0;
        for (int j = i + 1; j < 8; j++)
        {
            if (t.a[j] < t.a[i])
                tmp++;
        }
        num+=tmp*fac[7 - i];
    }
    return num;
}
int main()
{
    int i;
```

```
    for (i=0;i<8;i++)
        scanf("%d",&u.a[i]);
    printf("%d\n",contor(u));
}
```

22、Splay tree

```
#include<cstdio>
#define N 200000
#define INF 2000000000
int s[N][3],f[N],cnt[N],data[N],root,total;//son1left 2right
int le[N],m,x,n,i,j,ans;
char opt;
void rotate(int x,int w)//w:1leftZag 2rightZig
{
    int y;
    y=f[x];
    cnt[y]=cnt[y]-cnt[x]+cnt[s[x][w]];
    cnt[x]=cnt[x]-cnt[s[x][w]]+cnt[y];
    s[y][3-w]=s[x][w];
    if (s[x][w]!=0)
        f[s[x][w]]=y;
    f[x]=f[y];
    if (f[y]!=0)
        if (y==s[f[y]][1])
            s[f[y]][1]=x;
        else
            s[f[y]][2]=x;
    f[y]=x;
    s[x][w]=y;
}
void splay(int x,int t)
{
    int y;
    while (f[x]!=t)
    {
        y=f[x];
        if (f[y]==t)
            if (x==s[y][1])
                rotate(x,2);
            else
                rotate(x,1);
        else
            if (y==s[f[y]][1])
                if (x==s[y][1])
                {
                    rotate(y,2);
                    rotate(x,2);
                }
                else
                {
                    rotate(x,1);
                    rotate(x,2);
```

```
47              }
48          else
49              if (x==s[y][2])
50              {
51                  rotate(y,1);
52                  rotate(x,1);
53              }
54              else
55              {
56                  rotate(x,2);
57                  rotate(x,1);
58              }
59      }
60      if (f[x]==0) root=x;
61  }
62
63  int sea(int x,int w)
64  {
65      int t;
66      t=x;
67      while (1)
68      {
69          if (w<=data[t])
70          {
71              if (s[t][1]==0) break;
72              t=s[t][1];
73          }
74          else
75          {
76              if (s[t][2]==0) break;
77              t=s[t][2];
78          }
79          //if (data[t]==w) break;
80      }
81      splay(t,0);
82      return t;
83  }
84  void add(int w)
85  {
86      int x;
87      if (total==0)
88      {
89          total++;
90          f[1]=0;cnt[1]=1;data[1]=w;root=1;
91          return;
92      }
93      x=root;
94      while(1)
95      {
96          cnt[x]++;
97          if (w<=data[x])
98          {
99              if (s[x][1]==0) break;
100             x=s[x][1];
101         }
102         else
103         {
104             if (s[x][2]==0) break;
```

```
            x=s[x][2];
        }
    }
    total++;
    data[total]=w;
    f[total]=x;
    cnt[total]=1;
    if (w<=data[x])
        s[x][1]=total;
    else
        s[x][2]=total;
    splay(total,0);
}

int extract(int x,int w)//w=1min w=2max
{
    int i=x;
    if (w==1)
        i=sea(x,-INF);
    else
        i=sea(x,INF);
    splay(i,0);
    return i;
}
void clear(int x)
{
    cnt[x]=s[x][1]=s[x][2]=f[x]=data[x]=0;
}
void del(int w)
{
    int i;
    splay(w,0);
    i=w;
    if (s[i][1]==0)
    {
        root=s[i][2];
        f[root]=0;
    }
    if (s[i][2]==0)
    {
        root=s[i][1];
        f[root]=0;
    }
    if (s[i][2]!=0 && s[i][1]!=0)
    {
        f[s[i][1]]=0;
        root=extract(s[i][1],2);
        s[root][2]=s[i][2];
        cnt[root]+=cnt[s[i][2]];
        f[s[root][2]]=root;
    }
    clear(i);
}
int find(int k,int w)//w=1 Kthsmall;w=2 Kthbig
{
    int i,t;
    i=root;t=k;
    while (t!=cnt[s[i][w]]+1)
```

```
    {
        if (t>cnt[s[i][w]]+1)
        {
            t-=cnt[s[i][w]]+1;
            i=s[i][3-w];
        }
        else
            i=s[i][w];
    }
    splay(i,0);
    return i;
}
int main()
{
    scanf("%d%d",&n,&m);
    add(10000000);
    for (i=1;i<=n;i++)
    {
        getchar();
        scanf("%c %d",&opt,&x);
        if (opt=='I')
            if (x>=m)
                add(x);
        if (opt=='A')
        {
            for (j=1;j<=total;j++)
                data[j]+=x;
        }
        if (opt=='F')
        {
            x++;
            if (total-ans<x)
                printf("-1\n");
            else
                printf("%d\n",data[find(x,2)]);
        }
        if (opt=='S')
        {
            for (j=1;j<=total;j++)
                data[j]-=x;
            for (j=1;j<=total;j++)
                if (data[j]<m && le[j]==0)
                {
                    le[j]=1;
                    ans++;
                    del(j);
                }
        }
    }
    printf("%d\n",ans);
    return 0;
}
```

23、快速排序

```
void sort(int l,int r)
{
    int i,j,x;
    i=l;j=r;x=a[(l+r)/2];
    while (i<=j)
    {
        while (a[i]<x) i++;
        while (a[j]>x) j--;
        if (i<=j)
        {
            swap(a[i],a[j]);
            i++;j--;
        }
    }
    if (l<j) sort(l,j);
    if (i<r) sort(i,r);
}
```

24、头文件

```
#include <map>
#include <set>
#include <stack>
#include <queue>
#include <cmath>
#include <ctime>
#include <vector>
#include <cstdio>
#include <cctype>
#include <cstring>
#include <cstdlib>
#include <iostream>
#include <algorithm>
#define ll long long
#define ull unsigned long long
#define all(x) (x).begin(), (x).end()
#define ayacin ios::sync_with_stdio(false);
#define yukari 10000000000000000LL;
#define mod 1000000007
#define M_PI12 3.141592653
#define Parsee long double
#define reimu iterator
#define x first
#define y second
#define mokou __builtin_popcount
#define __sum accumulate
#define rep(i,a,b) for(i=a;i<=b;i++)
#define lson l , m , rt << 1
#define rson m + 1 , r , rt << 1 | 1
#define eps 1e-10
```

```
#define zero(a) fabs(a)<eps

struct qwe
{
    int x,c;
    bool operator < (const qwe &t) const
    {
        return x>t.x;
    }
};
struct wer
{
    int x,y,c;
}e[5000001];
priority_queue<qwe>que;
```

25、bsgs

```
#include<cstdio>
#include<cmath>
#include<iostream>
#include<cstring>
#include<algorithm>
#include<cmath>
#include<map>
#include<set>
using namespace std;
#define ll   int64
#define usint unsigned int
#define RE  freopen("1.in","r",stdin)

class hash {
public:
    hash() {
        memset(a,0xff,sizeof(a));
    }
    int locate(ll x) {
        ll l=x%MOD;
        while(a[l]!=x&&a[l]!=-1) l=(l+1)%MOD;
        return l;
    }
    void insert(ll x,ll va) {
        ll l=locate(x);
        if(a[l]==-1) {
            a[l]=x;
            v[l]=va;
        }
    }
    ll get(ll x) {
        ll l=locate(x);
        return a[l]==x?v[l]:-1;
    }
```

```
    void clear() {
        memset(a,0xff,sizeof(a));
    }
private:
    static const ll MOD=100007;
    ll a[MOD+100],v[MOD+100];
} S;

ll exgcd(ll a,ll b,ll &x,ll &y) {
    ll t,ret;
    if (!b) {
        x=1,y=0;
        return a;
    }
    ret=exgcd(b,a%b,x,y);
    t=x,x=y,y=t-a/b*y;
    return ret;
}

int main() {///A^x=B(modC)    A^j=B*A^(-m*i)(mod C)
    ll C,A,B;
    ll m,i,t,D,ans,x,y;
    while(scanf("%lld%lld%lld",&C,&A,&B)!=EOF) {
        S.clear();
        m=ceil(sqrt((double)C));
        t=1;
        for(i=0; i<m; i++) {
            S.insert(t,i);
            t=t*A%C;
        }
        D=1;///此时 t=A^m
        ans=-1;
        for(i=0; i<m; i++) {
            exgcd(D,C,x,y);///exgcd 求逆元，得到 x=D^(-i*m)
            x=((x*B)%C+C)%C;///B*x=B*D^(-i*m)
            y=S.get(x);
            //printf("%lld,%lld\n",x,y);
            if(y!=-1) {
                ans=i*m+y;
                break;
            }
            D=(D*t)%C;///D=t^i,(t=A^m)
        }
        if(ans==-1) printf("no solution\n");
        else printf("%lld\n",ans);
    }
    return 0;
}
```

26、莫队算法

```
#include <bits/stdc++.h>
#define fi first
```

```cpp
#define se second
#define mp make pair
#define pb push back
using namespace std;
const int N = 50010;
struct Query
{
    int L,R,id;
} node[N];
int ans[N],a[N],b[N],n,m,k,unit,i;
pair <int,int> t[1000010];
bool cmp(Query a,Query b)
{
    if(a.L/unit != b.L/unit)return a.L/unit < b.L/unit;
    else return a.R < b.R;
}
void solve()
{
    int temp = 0;
    memset(t,0,sizeof(t));
    int L = 1;
    int R = 0;
    for(int i = 1; i <=m; i++)
    {
        while(R < node[i].R)
        {
            R++;
            if (t[a[R]].fi<t[a[R]].se) temp++;
            t[a[R]].fi++;
            if (t[b[R]].se<t[b[R]].fi) temp++;
            t[b[R]].se++;
        }
        while(R > node[i].R)
        {
            if (t[a[R]].fi<=t[a[R]].se) temp--;
            t[a[R]].fi--;
            if (t[b[R]].se<=t[b[R]].fi) temp--;
            t[b[R]].se--;
            R--;
        }
        while(L < node[i].L)
        {
            if (t[a[L]].fi<=t[a[L]].se) temp--;
            t[a[L]].fi--;
            if (t[b[L]].se<=t[b[L]].fi) temp--;
            t[b[L]].se--;
            L++;
        }
        while(L > node[i].L)
        {
            L--;
            if (t[a[L]].fi<t[a[L]].se) temp++;
            t[a[L]].fi++;
            if (t[b[L]].se<t[b[L]].fi) temp++;
            t[b[L]].se++;
        }
        ans[node[i].id]=temp;
    }
```

```
}
int main()
{
    scanf("%d%d%d",&n,&m,&k);
    for (i=1; i<=n; i++) scanf("%d",&a[i]);
    for (i=1; i<=n; i++) scanf("%d",&b[i]);
    for(i = 1; i<=m; i++)
    {
        node[i].id = i;
        scanf("%d%d",&node[i].L,&node[i].R);
        node[i].L++,node[i].R++;
    }
    unit = (int)sqrt(n);
    sort(node+1,node+m+1,cmp);
    solve();
    for (i=1; i<=m; i++)
        printf("%d\n",ans[i]);
    return 0;
}
/*
3 4 2
0 0 1
0 0 0
0 0
2 2
0 1
1 2

7 10 3
0 1 2 1 1 0 2
1 0 0 1 0 2 1
*/
```

27、treap

```
struct Node{
    Node *s[2];
    int v, w, pos, mk, cnt;
    //ll r[5], p;
    //ll getv(int i = 0){ return r[(p % 5 + i + 5) % 5]; }
    Node *link(int w, Node *p){ s[w] = p; return this; }
    void init(){ memset(this, 0, sizeof(*this)); w = rand(); cnt = 1; }
    void push();
    void update();
};
struct   HNIUNO{
    static const int size = 1e6 + 5;
    Node spc[size];
    int p;
    void reset(){ p = 0; }
    Node *nloc(){ spc[p].init(); return spc + (p++); }
}nodes;
```

```
18 void rot(Node *&p, int lr){
19     Node *q = p->s[!lr];
20     p->push();
21     q->push();
22     lr ? (q->link(1, p->link(0, q->s[1]))) : (q->link(0, p->link(1,
   q->s[0])));
23     p->update();
24     q->update();
25     p = q;
26 }
27 void insert(Node *&p, int v, int pos = 1){
28     if (p == NULL){ p = nodes.nloc(); p->v = v; p->pos = pos; p->update();
   return; }
29     if (p->v == v)return;
30     p->push();
31     if (v < p->v){
32         insert(p->s[0], v, pos);
33         //p->mk++;
34         //p->s[0]->mk--;
35     }
36     else{
37         insert(p->s[1], v, pos + 1 + (p->s[0] ? p->s[0]->cnt : 0));
38     }
39     p->update();
40     int w = v > p->v;
41     if (p->s[w]->w < p->w)rot(p, !w);
42 }
43 void del(Node *&p){
44     p->push();
45     if (p->s[0] == NULL){ p = p->s[1]; return; }
46     if (p->s[1] == NULL){ p = p->s[0]; return; }
47     p->s[0]->w < p->s[1]->w ? (rot(p, 1), del(p->s[1]), p->update()) :
   (rot(p, 0), del(p->s[0]), p->update());
48 }
49 void del(Node *&p, int v){
50     if (p == NULL)return;
51     p->push();
52     if (p->v == v){ if (p->s[1])p->s[1]->mk--; del(p); }
53     else{
54         if (v < p->v){
55             //p->mk--;
56             //if (p->s[0]) p->s[0]->mk++;
57             del(p->s[0], v);
58         }
59         else{
60             del(p->s[1], v);
61         }
62         p->update();
63     }
64 }
```

28、NTT

```
typedef long long LL;
const int maxn=200000;
const LL P=1004535809; //P=C*2^k+1，P 是质数
const LL g=3; //P 的原根
int N,na,nb;
int a[maxn*2],b[maxn*2],W[2][maxn*2],rev[maxn*2];

LL Pow(LL a,int b)
{
    LL c=1;
    for (;b; b>>=1,a=a*a%P) if (b&1) c=c*a%P;
    return c;
}
void NTT(int*a,int f)
{
    for (int i=0;i<N;i++) if (i<rev[i]) swap(a[i],a[rev[i]]);
    for (int i=1;i<N;i<<=1)
        for (int j=0,t=N/(i<<1);j<N;j+=i<<1)
            for (int k=0,l=0,x,y;k<i;k++,l+=t)
                x=(LL)W[f][l]*a[j+k+i]%P,y=a[j+k],a[j+k]=(y+x)%P,a[j+k+i]=(y-
x+P)%P;
    if (f) for (int i=0,x=Pow(N,P-2);i<N;i++) a[i]=(LL)a[i]*x%P;
}
void change()
{
    for (int i=0;i<N;i++)
    {
        int x=i,y=0;
        for (int k=1; k<N; x>>=1,k<<=1) (y<<=1)|=x&1;
        rev[i]=y;
    }
}
int main()
{
    for (N=1; N<na||N<nb; N<<=1); N<<=1;
    for (int i=na;i<N;i++)
      a[i]=b[i]=0;
    W[0][0]=W[1][0]=1;
    for (int i=1,x=Pow(g,(P-1)/N),y=Pow(x,P-2); i<N; i++)
        W[0][i]=(LL)x*W[0][i-1]%P,W[1][i]=(LL)y*W[1][i-1]%P;
    change();
    NTT(a,0),NTT(b,0);
    for (int i=0;i<N;i++) a[i]=(LL)a[i]*b[i]%P;
    NTT(a,1);
}
```

29、Java 大数 a+b

```
import java.math.BigInteger;
import java.util.Scanner;
```

```
3
4  public class Main{
5      public static void main(String[] args){
6          Scanner scan = new Scanner(System.in);
7          while(scan.hasNextInt()){
8              int ncase = scan.nextInt();
9              for(int i = 1; i <= ncase; i++){
10                 BigInteger a = scan.nextBigInteger();
11                 BigInteger b = scan.nextBigInteger();
12                 System.out.println("Case " + i + ":");
13                 System.out.println(a + " + " + b + " = " + a.add(b));
14                 if(i != ncase){
15                     System.out.println();
16                 }
17             }
18         }
19     }
20 }
```