

ACM 模板

from XDU_升华

目录

1、AC 自动机.....	1
2、KMP	3
3、Manacher.....	3
4、后缀数组	4
5、最大流	7
6、最小费用最大流	8
7、最小树形图-朱刘算法.....	10
8、Tarjan 强连通分量.....	12
9、Tarjan 求割边割点.....	13
10、无源汇的最小割-Stoer-Wagner 算法	14
11、倍增 LAC.....	15
12、树链剖分	17
13、FFT 快速卷积	20
14、Lucas 定理求组合数	23
15、欧拉筛法	23
16、高斯消元	24
17、树状数组	25
18、二维树状数组	26
19、小根堆	27
20、圆面积交	28
21、Contor 展开.....	28
22、Splay tree.....	29
23、快速排序	33
24、头文件	33
25、bsgs.....	34

26、莫队算法.....	35
27、treap.....	37
28、NTT.....	39
29、Java 大数 a+b.....	39

1、AC 自动机

```

1  #include <cstdio>
2  #include <algorithm>
3  #include <iostream>
4  #include <cstring>
5  #include <queue>
6  using namespace std;
7  struct Trie
8  {
9      int next[500010][26],fail[500010],end[500010];
10     int root,L;
11     int newnode()
12     {
13         for(int i = 0;i < 26;i++)
14             next[L][i] = -1;
15         end[L++] = 0;
16         return L-1;
17     }
18     void init()
19     {
20         L = 0;
21         root = newnode();
22     }
23     void insert(char buf[])
24     {
25         int len = strlen(buf);
26         int now = root;
27         for(int i = 0;i < len;i++)
28         {
29             if(next[now][buf[i]-'a'] == -1)
30                 next[now][buf[i]-'a'] = newnode();
31             now = next[now][buf[i]-'a'];
32         }
33         end[now]++;
34     }
35     void build()
36     {
37         queue<int>Q;
38         fail[root] = root;
39         for(int i = 0;i < 26;i++)
40             if(next[root][i] == -1)
41                 next[root][i] = root;
42             else
43                 {
44                     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         {

```

```

103         scanf("%s",buf);
104         ac.insert(buf);
105     }
106     ac.build();
107     scanf("%s",buf);
108     printf("%d\n",ac.query(buf));
109 }
110 return 0;
111 }

```

2、KMP

```

1  #include<cstdio>
2  #include<cstring>
3  # define M 110000
4  int i,j,ans,m,n,next[M];
5  char s[M],t[M];
6  int main()
7  {
8      ans=0;
9      scanf("%s",s+1);
10     scanf("%s",t+1);
11     n=strlen(s+1);m=strlen(t+1);
12     j=0;
13     for (i=2;i<=n;i++)
14     {
15         while (j>0 && s[j+1]!=s[i])
16             j=next[j];
17         if (s[j+1]==s[i]) j++;
18         next[i]=j;
19     }
20     j=0;
21     for (i=1;i<=m;i++)
22     {
23         while (j>0 && s[j+1]!=t[i])
24             j=next[j];
25         if (s[j+1]==t[i]) j++;
26         if (j==n)
27         {
28             ans++;j=next[j];
29         }
30         printf("%d\n",ans);
31     }
32     //printf("%d\n",ans);
33     return 0;
34 }

```

3、Manacher

```

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

```

```

4  int p[M <<1];
5  int i,n,id,maxl,maxid;
6  int min(int a,int b)
7  {
8      return a<b?a:b;
9  }
10 int main()
11 {
12     while (scanf("%s",&b[1])!=EOF)
13     {
14         maxl=maxid=0;
15         for (i=1;b[i]!='\0';i++)
16         {
17             a[(i<<1)]=b[i];
18             a[(i<<1)+1]='#';
19             printf("%d\n",i);
20         }
21         i--;
22         a[0]='?';a[1]='#';
23         n=(i<<1)+2;a[n+1]=0;
24         //a[0],a[n+1]为不同的原串中没有的字符
25         for (i=1;i<n;i++)
26         {
27             if (maxid>i)
28                 p[i]=min(p[2*id-i],maxid-i);
29             else
30                 p[i]=1;
31             while (a[i+p[i]]==a[i-p[i]])
32                 p[i]++;
33             if (p[i]+i>maxid)
34             {
35                 maxid=p[i]+i;
36                 id=i;
37             }
38             if (p[i]>maxl) maxl=p[i];
39         }
40         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
41     }
42     return 0;
43 }

```

4、后缀数组

```

1  #include<cstdio>
2  #include<cstring>
3  #define N 1000001
4  int wa[N],wb[N],wv[N],ws[N];
5  int cmp(int *r,int a,int b,int l)
6  {return r[a]==r[b]&&r[a+l]==r[b+l];}
7  void da(int *r,int *sa,int n,int m)
8  {
9      int i,j,p,*x=wa,*y=wb,*t;

```

```

10     for(i=0;i<m;i++) ws[i]=0;
11     for(i=0;i<n;i++) ws[x[i]=r[i]]++;
12     for(i=1;i<m;i++) ws[i]+=ws[i-1];
13     for(i=n-1;i>=0;i--) sa[--ws[x[i]]]=i;
14     for(j=1,p=1;p<n;j*=2,m=p)
15     {
16         for(p=0,i=n-j;i<n;i++) y[p++]=i;
17         for(i=0;i<n;i++) if(sa[i]>=j) y[p++]=sa[i]-j;
18         for(i=0;i<n;i++) wv[i]=x[y[i]];
19         for(i=0;i<m;i++) ws[i]=0;
20         for(i=0;i<n;i++) ws[wv[i]]++;
21         for(i=1;i<m;i++) ws[i]+=ws[i-1];
22         for(i=n-1;i>=0;i--) sa[--ws[wv[i]]]=y[i];
23         for(t=x,x=y,y=t,p=1,x[sa[0]]=0,i=1;i<n;i++)
24             x[sa[i]]=cmp(y,sa[i-1],sa[i],j)?p-1:p++;
25     }
26     return;
27 }
28 /* DC3TTCME
29 #define N 1000003
30 #define F(x) ((x)/3+((x)%3==1?0:tb))
31 #define G(x) ((x)<tb?(x)*3+1:(x)-tb)*3+2)
32 int wa[N],wb[N],wv[N],ws[N];
33 int c0(int *r,int a,int b)
34 {return r[a]==r[b]&&r[a+1]==r[b+1]&&r[a+2]==r[b+2];}
35 int c12(int k,int *r,int a,int b)
36 {if(k==2) return r[a]<r[b]||r[a]==r[b]&&c12(1,r,a+1,b+1);
37  else return r[a]<r[b]||r[a]==r[b]&&wv[a+1]<wv[b+1];}
38 void sort(int *r,int *a,int *b,int n,int m)
39 {
40     int i;
41     for(i=0;i<n;i++) wv[i]=r[a[i]];
42     for(i=0;i<m;i++) ws[i]=0;
43     for(i=0;i<n;i++) ws[wv[i]]++;
44     for(i=1;i<m;i++) ws[i]+=ws[i-1];
45     for(i=n-1;i>=0;i--) b[--ws[wv[i]]]=a[i];
46     return;
47 }
48 void dc3(int *r,int *sa,int n,int m)
49 {
50     int i,j,*rn=r+n,*san=sa+n,ta=0,tb=(n+1)/3,tbc=0,p;
51     r[n]=r[n+1]=0;
52     for(i=0;i<n;i++) if(i%3!=0) wa[tbc++]=i;
53     sort(r+2,wa,wb,tbc,m);
54     sort(r+1,wb,wa,tbc,m);
55     sort(r,wa,wb,tbc,m);
56     for(p=1,rn[F(wb[0])]=0,i=1;i<tbc;i++)
57         rn[F(wb[i])]=c0(r,wb[i-1],wb[i])?p-1:p++;
58     if(p<tbc) dc3(rn,san,tbc,p);
59     else for(i=0;i<tbc;i++) san[rn[i]]=i;
60     for(i=0;i<tbc;i++) if(san[i]<tb) wb[ta++]=san[i]*3;
61     if(n%3==1) wb[ta++]=n-1;
62     sort(r,wb,wa,ta,m);
63     for(i=0;i<tbc;i++) wv[wb[i]=G(san[i])]=i;
64     for(i=0,j=0,p=0;i<ta && j<tbc;p++)
65         sa[p]=c12(wb[j]%3,r,wa[i],wb[j])?wa[i++]:wb[j++];
66     for(;i<ta;p++) sa[p]=wa[i++];
67     for(;j<tbc;p++) sa[p]=wb[j++];

```

```

68     return;
69 }
70 */
71 int rank[N],height[N];
72 void calheight(int *r,int *sa,int n)
73 {
74     int i,j,k=0;
75     for(i=1;i<=n;i++) rank[sa[i]]=i;
76     for(i=0;i<n;height[rank[i++]]=k)
77         for(k?k--:0,j=sa[rank[i]-1];r[i+k]==r[j+k];k++);
78     return;
79 }
80 int RMQ[N];
81 int mm[N];
82 int best[20][N];
83 void initRMQ(int n)
84 {
85     int i,j,a,b;
86     for(mm[0]=-1,i=1;i<=n;i++)
87         mm[i]=((i&(i-1))==0)?mm[i-1]+1:mm[i-1];
88     for(i=1;i<=n;i++) best[0][i]=i;
89     for(i=1;i<=mm[n];i++)
90         for(j=1;j<=n+1-(1<<i);j++)
91         {
92             a=best[i-1][j];
93             b=best[i-1][j+(1<<(i-1))];
94             if(RMQ[a]<RMQ[b]) best[i][j]=a;
95             else best[i][j]=b;
96         }
97     return;
98 }
99 int askRMQ(int a,int b)
100 {
101     int t;
102     t=mm[b-a+1];b-=(1<<t)-1;
103     a=best[t][a];b=best[t][b];
104     return RMQ[a]<RMQ[b]?a:b;
105 }
106 int lcp(int a,int b)
107 {
108     int t;
109     a=rank[a];b=rank[b];
110     if(a>b) {t=a;a=b;b=t;}
111     return(height[askRMQ(a+1,b)]);
112 }
113 int ans, ,n,i,a[N],sa[N];
114 char s[N];
115 void debug()
116 {
117     int i;
118     for (i=0;i<n;i++)
119         printf("%d ",sa[i]);
120     printf("%d\n",sa[n]);
121     for (i=0;i<n;i++)
122         printf("%d ",rank[i]);
123     printf("%d\n",rank[n]);
124     for (i=0;i<n;i++)
125         printf("%d ",height[i]);

```

```

126     printf("%d\n",height[i]);
127 }
128
129 int main()
130 {
131     scanf("%d",& );
132     while ( --)
133     {
134         ans=0;
135         scanf("%s",s);
136         n=strlen(s);
137         for (i=0;i<n;i++)
138             a[i]=s[i];
139         a[n]=0;
140         da(a,sa,n+1,300);
141         calheight(a,sa,n);
142         debug();
143         for (i=1;i<=n;i++)
144             ans+=n-sa[i]-height[i];
145         printf("%d\n",ans);
146     }
147     return 0;
148 }

```

5、最大流

```

1  #include<cstdio>
2  #include<cstring>
3  #define M 210
4  #define N 210
5  int len[M <<1],e[M <<1],nex[M <<1],other[M
6  <<1],head[N],last[N],d[N],num[N];
7  int m,n,i,ans,tot,ss,tt,ee,u,v,c;
8  int init()
9  {
10     memset(head,0,sizeof(head));
11     memset(num,0,sizeof(num));
12     memset(d,0,sizeof(d));
13     ans=ee=0;
14 }
15 int min(int a,int b)
16 {
17     return a<b?a:b;
18 }
19 void add(int u,int v,int c)
20 {
21     other[++ee]=ee+1;
22     e[ee]=v;nex[ee]=head[u];head[u]=ee;len[ee]=c;
23     other[++ee]=ee-1;
24     e[ee]=u;nex[ee]=head[v];head[v]=ee;len[ee]=0;
25 }
26 int dfs(int x,int flow)
27 {

```



```

28     int rec,j,p;
29     if (x==tt) return flow;
30     rec=0;j=last[x];
31     while (j!=0)
32     {
33         if (len[j]>0 && d[x]==d[e[j]]+1)
34         {
35             last[x]=j;
36             p=dfs(e[j],min(len[j],flow-rec));
37             len[j]-=p;len[other[j]]+=p;
38             rec+=p;
39             if (rec==flow) return rec;
40         }
41         j=nex[j];
42     }
43     if (d[ss]>tot) return rec;
44     if (--num[d[x]]==0) d[ss]=tot;
45     last[x]=head[x];
46     num[++d[x]]++;
47     return rec;
48 }
49 int main()
50 {
51     while (scanf("%d%d",&m,&n)!=EOF)
52     {
53         init();
54         for (i=1;i<=m;i++)
55         {
56             scanf("%d%d%d",&u,&v,&c);
57             add(u,v,c);
58         }
59         tot=num[0]=n;
60         ss=1;tt=n;
61         for (i=ss;i<=tt;i++)
62             last[i]=head[i];
63         while (d[ss]<tot)
64             ans+=dfs(ss,2147483647);
65         printf("%d\n",ans);
66     }
67     return 0;
68 }

```

6、最小费用最大流

```

1  #include<cstdio>
2  #include<cstring>
3  #define INF 1000000000
4  #define N 1010
5  #define M 30010
6  #define WW 10000
7  int _,other[M],e[M],x[M],nex[M],len[M],ww[M];
8

```

```

9   int
10  pre[N],vis[N],head[N],dis[N],a[500000],ss,tt,n,m,i,j,k,opt,maxflow,mincos
11  t,ee;
12  int xx,xq[300][300],gy[300][300],sum,summ,flag;
13  int min(int a,int b)
14  {return a<b?a:b;}
15  bool spfa()
16  {
17      int h,t,i,j;
18      h=1;t=1;
19      for (i=0;i<=tt;i++) dis[i]=INF;
20      a[1]=ss;pre[ss]=-1;
21      vis[ss]=1;dis[ss]=0;
22      while (h<=t)
23      {
24          j=head[a[h]];
25          while (j!=0)
26          {
27              if (len[j]>0 && dis[a[h]]+ww[j]<dis[e[j]])
28              {
29                  dis[e[j]]=dis[a[h]]+ww[j];
30                  pre[e[j]]=j;
31                  if (vis[e[j]]==0)
32                  {
33                      vis[e[j]]=1;a[++t]=e[j];
34                  }
35              }
36              j=nex[j];
37          }
38          vis[a[h++]]=0;
39      }
40      if (dis[tt]>=INF-1000) return 0;
41      else return 1;
42  }
43
44  void work()
45  {
46      int zh,flow;
47      mincost=maxflow=0;
48      while (spfa())
49      {
50          zh=tt;
51          flow=INF;
52          while (pre[zh]!=-1)
53          {
54              flow=min(flow,len[pre[zh]]);
55              zh=x[pre[zh]];
56          }
57          maxflow+=flow;
58          mincost+=dis[tt]*flow;
59          zh=tt;
60          while (pre[zh]!=-1)
61          {
62              len[pre[zh]]-=flow;
63              len[other[pre[zh]]]+=flow;
64              zh=x[pre[zh]];
65          }

```

```

66     }
67 }
68
69 void add(int u,int v,int c,int w)
70 {
71     e[++ee]=v;x[ee]=u;nex[ee]=head[u];head[u]=ee;
72     len[ee]=c;ww[ee]=w;other[ee]=ee+1;
73     e[++ee]=u;x[ee]=v;nex[ee]=head[v];head[v]=ee;
74     len[ee]=0;ww[ee]=-w;other[ee]=ee-1;
75 }
76
77 void init()
78 {
79     memset(head,0,sizeof(head));
80     memset(nex,0,sizeof(nex));
81     sum=ee=0;
82 }
83 int main()
84 {
85     scanf("%d",& );
86     while ( --)
87     {
88         init();
89         n=3;
90         add(1,2,1,-1);
91         add(2,3,1,-1);
92         add(3,1,1,-1);
93         ss=1;tt=n;
94         work();
95     }
96     return 0;
97 }

```

7、最小树形图-朱刘算法

```

1  #include <cstdio>
2  #include <iostream>
3  #include <cmath>
4  #include <cstring>
5  using namespace std;
6  const int N=101,M=10001,inf=2147483647;
7  struct edge {int u,v;double w;} e[M];
8  int i,n,m,id[N],pre[N],v[N];
9  double x[N],y[N],ans,inw[N];
10 void zhu liu(int root)
11 {
12     int s,t,idx=1;
13     while (idx)
14     {
15         for (int i=1;i<=n;++i) inw[i]=inf,id[i]=0,v[i]=0;
16         for (int i=1;i<=m;++i)
17         {
18             s=e[i].u;t=e[i].v;

```

```

19         if (e[i].w>=inw[t] || s==t) continue;
20         pre[t]=s;
21         inw[t]=e[i].w;
22     }
23     //for (i=1;i<=n;i++)printf("%f\n",inw[i]);
24     inw[root]=0;pre[root]=root;
25     for (int i=1;i<=n;++i)
26     {
27         if (inw[i]==inf)
28         {
29             printf("poor snoopy\n");
30             return;
31         }
32         ans+=inw[i];
33     }
34     idx=0;
35     for (int i=1;i<=n;++i)
36         if (v[i]==0)
37         {
38             t=i;
39             while (v[t]==0) v[t]=i,t=pre[t];
40             if (v[t]!=i || t==root) continue;
41             id[t]=++idx;
42             for (s=pre[t];s!=t;s=pre[s]) id[s]=idx;
43         }
44     if (idx==0) continue;
45     for (int i=1;i<=n;++i)
46         if (id[i]==0) id[i]=++idx;
47     for (int i=1;i<=m;++i)
48     {
49         e[i].w-=inw[e[i].v];
50         e[i].u=id[e[i].u];
51         e[i].v=id[e[i].v];
52     }
53     n=idx;
54     root=id[root];
55 }
56 printf("%.2f\n",ans);
57 }
58 int main()
59 {
60     while (scanf("%d%d",&n,&m)==2)
61     {
62         ans=0;
63         for (i=1;i<=n;i++)
64             scanf("%lf%lf",&x[i],&y[i]);
65         for (i=1;i<=m;i++)
66         {
67             scanf("%d%d",&e[i].u,&e[i].v);
68             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]));
69         }
70         zhu liu(1);
71     }
72     return 0;
73 }

```

8、Tarjan 强连通分量

```
1  #include<cstdio>
2  #include<cstring>
3  #define M 100001
4  #define N 100001
5  int stap[N],e[M],next[M],dfn[N],head[N],low[N],c[N];
6  bool instap[N];
7  int color,time,stop,i,n,m,u,v,ee;
8  int min(int a,int b)
9  {
10     return a<b?a:b;
11 }
12 void tarjan(int u)
13 {
14     int j;
15     dfn[u]=low[u]=++time;
16     instap[u]=1;
17     stap[++stop]=u;
18     for (j=head[u];j;j=next[j])
19     {
20         if (!dfn[e[j]])
21         {
22             tarjan(e[j]);
23             low[u]=min(low[u],low[e[j]]);
24         }
25         else if (instap[e[j]])
26             low[u]=min(low[u],dfn[e[j]]);
27     }
28     if (dfn[u]==low[u])
29     {
30         color++;
31         while (1)
32         {
33             j=stap[stop--];
34             instap[j]=0;
35             c[j]=color;
36             if (j==u) break;
37         }
38     }
39 }
40 void add(int u,int v)
41 {
42     e[++ee]=v;next[ee]=head[u];head[u]=ee;
43 }
44 void init()
45 {
46     ee=stop=color=time=0;
47     memset(head,0,sizeof(head));
48     memset(next,0,sizeof(next));
49     memset(dfn,0,sizeof(dfn));
50     memset(instap,0,sizeof(instap));
51 }
52 int main()
53 {
54     while (scanf("%d%d",&n,&m)==2)
```

```

55     {
56         init();
57         for (i=1;i<=m;i++)
58         {
59             scanf("%d%d",&u,&v);
60             add(u,v);
61         }
62         for (i=1;i<=n;i++)
63             if (!dfn[i]) tarjan(i);
64         for (i=1;i<=n;i++)
65             printf("%d ",c[i]);
66     }
67     return 0;
68 }

```

9、Tarjan 求割边割点

```

1  /*if (ll[e[i]]>pre[u]) i 是割边*/
2  #include<cstdio>
3  #include<cstring>
4  #define M 100001
5  #define N 100001
6  int e[M],next[M],dfn[N],head[N],low[N];
7  int ee,du,i,n,m,u,v,time;
8  bool vis[M];
9  inline int min(int a,int b)
10 {
11     return a<b?a:b;
12 }
13 void tarjan(int u)
14 {
15     int j;
16     low[u]=dfn[u]=++time;
17     for (j=head[u];j;j=next[j])
18         if (!vis[j])
19         {
20             vis[j]=1;vis[((j+1)^1)-1]=1;
21             if (dfn[e[j]]==0)
22             {
23                 tarjan(e[j]);
24                 low[u]=min(low[u],low[e[j]]);
25                 if (u==1) du++;
26                 if (low[e[j]]>=dfn[u]) printf("%d\n",u);
27             }
28             else low[u]=min(low[u],dfn[e[j]]);
29         }
30 }
31 void add(int u,int v)
32 {
33     e[++ee]=v;next[ee]=head[u];head[u]=ee;
34     e[++ee]=u;next[ee]=head[v];head[v]=ee;
35 }
36 void init()

```

```

37 {
38     time=du=ee=0;
39     memset(head,0,sizeof(head));
40     memset(next,0,sizeof(next));
41     memset(dfn,0,sizeof(dfn));
42     memset(vis,0,sizeof(vis));
43     memset(low,0,sizeof(low));
44 }
45 int main()
46 {
47     while (scanf("%d%d",&n,&m)==2)
48     {
49         init();
50         for (i=1;i<=m;i++)
51         {
52             scanf("%d%d",&u,&v);
53             add(u,v);
54         }
55         tarjan(1);
56     }
57     return 0;
58 }

```

10、无源汇的最小割-Stoer-Wagner 算法

```

1  #include <cstdio>
2  #include <cstring>
3  #define INF 1000000000
4  #define N 510
5  using namespace std;
6
7  int mp[N][N];
8  int i,n,m,u,v,c;
9  bool combine[N];
10 int minC=INF;
11
12 void search(int &s,int &t)
13 {
14     bool vis[N];
15     int i,j,w[N];
16     memset(vis,0,sizeof(vis));
17     memset(w,0,sizeof(w));
18     int tmpj=1000;
19     for(i=1;i<=n;i++)
20     {
21         int max=-INF;
22         for(j=1;j<=n;j++)
23         {
24             if(!vis[j]&&!combine[j]&&max<w[j])
25             {
26                 max=w[j];
27                 tmpj=j;
28             }

```

```

29     }
30     if(t==tmpj){minC=w[t];return;}
31     vis[tmpj]=1;
32     s=t,t=tmpj;
33     for(int j=1;j<=n;j++)
34     {
35         if(!vis[j]&&!combine[j])
36             w[j]+=mp[t][j];
37     }
38 }
39 minC=w[t];
40 }
41
42 int mincut()
43 {
44     int ans=INF,i,j,s,t;
45     memset(combine,0,sizeof(combine));
46     for(i=1;i<=n-1;i++)
47     {
48         s=t=-1;
49         search(s,t);
50         combine[t]=true;
51         ans=ans>minC?minC:ans;
52         for(j=1;j<=n;j++)
53         {
54             mp[s][j]+=mp[t][j];
55             mp[j][s]+=mp[j][t];
56         }
57     }
58     return ans;
59 }
60 int main()
61 {
62     while(scanf("%d%d",&n,&m)==2)
63     {
64         memset(mp,0,sizeof(mp));
65         for(i=1;i<=m;i++)
66         {
67             scanf("%d%d%d",&u,&v,&c);
68             u++;v++;
69             mp[u][v]+=c;
70             mp[v][u]+=c;
71         }
72         printf("%d\n",mincut());
73     }
74     return 0;
75 }

```

11、倍增 LAC

```

1  #include<cstdio>
2  #include<cstring>
3  #include<algorithm>

```



```

4  #include<cmath>
5  #include<string>
6  #include<iostream>
7  #include<map>
8  #define MAX(a,b) a>b?a:b
9  #define N 110000
10 #define mem(a) memset(a,0,sizeof(a))
11 using namespace std;
12 int l[N<<1],sum,j,k,deep,hh,tt,ee,i,n,ans,e[N<<1],head[N],nex[N<<1];
13 int q,f[N],a[N<<2],u,v,c,dep[N],len[N][20],p[N][20];
14 void add(int u,int v,int c)
15 {
16     e[++ee]=v;nex[ee]=head[u];head[u]=ee;l[ee]=c;
17 }
18 int lca(int a,int b)
19 {
20     int i,j,k;
21     if (dep[a]>dep[b])
22         swap(a,b);
23     k=(int)(log(dep[b])/log(2));
24     for (i=k;i>=0;i--)
25         if (dep[b]-(1<<i)>=dep[a])
26         {
27             sum+=len[b][i];
28             b=p[b][i];
29         }
30     if (a==b) return a;
31     k=(int)(log(dep[b])/log(2));
32     for (i=k;i>=0;i--)
33         if (p[a][i]!=p[b][i])
34         {
35             sum+=len[b][i]+len[a][i];
36             a=p[a][i];
37             b=p[b][i];
38         }
39     sum+=len[a][0]+len[b][0];
40     return f[a];
41 }
42
43 int main()
44 {
45     while (scanf("%d%d",&n,&q)==2)
46     {
47         mem(head);mem(dep);mem(p);
48         ee=0;
49         for (i=1;i<=n-1;i++)
50         {
51             scanf("%d%d%d",&u,&v,&c);
52             add(u,v,c);
53             add(v,u,c);
54         }
55         a[1]=1;hh=tt=1;dep[1]=1;
56         while (hh<=tt)
57         {
58             j=head[a[hh]];
59             while (j>0)
60             {
61                 if (dep[e[j]]==0)

```

```

62         {
63             dep[e[j]]=dep[a[hh]]+1;
64             deep=MAX(deep,dep[e[j]]);
65             f[e[j]]=a[hh];
66             len[e[j]][0]=l[j];
67             a[++tt]=e[j];
68         }
69         j=nex[j];
70     }
71     hh++;
72 }
73 for (i=1;i<=n;i++)
74     p[i][0]=f[i];
75 k=(int)(log(deep)/log(2));
76 for (j=1;j<=k;j++)
77     for (i=1;i<=n;i++)
78     {
79         p[i][j]=p[p[i][j-1]][j-1];
80         len[i][j]=len[i][j-1]+len[p[i][j-1]][j-1];
81     }
82 }
83 return 0;
84 }

```

12、树链剖分

```

1  #include<cstdio>
2  #include<cstring>
3  #include<algorithm>
4  #define MAX(a,b) a>b?a:b
5  #define N 110000
6  #define mem(a) memset(a,0,sizeof(a))
7  using namespace std;
8  int
9  uu,vv,ma,root,ee,rr,ll, ,i,n,tot,ans,e[N<<1],head[N],len[N<<1],nex[N<<1];
10 int
11 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];
12 char opt[100];
13 void add(int u,int v,int c)
14 {
15     e[++ee]=v;nex[ee]=head[u];head[u]=ee;
16 }
17 void build(int s,int ll,int rr)
18 {
19     l[s]=ll;r[s]=rr;
20     if (ll==rr)
21         a[s]=0;
22     else
23     {
24         int mid=(ll+rr)>>1;
25         build(s<<1,ll,mid);
26         build((s<<1)+1,mid+1,rr);
27     }
28 }

```

```

26 }
27 void add(int s)
28 {
29     if (l[s]==r[s])
30         a[s]=rr;
31     else
32     {
33         if (r[s<<1]>=ll)
34             add(s<<1);
35         else
36             add((s<<1)+1);
37         a[s]=MAX(a[s<<1],a[(s<<1)+1]);
38     }
39 }
40 void sea(int s)
41 {
42     if (l[s]>rr || r[s]<ll)
43         return;
44     if (l[s]>=ll&&r[s]<=rr)
45         ans=MAX(ans,a[s]);
46     else
47     {
48         sea(s<<1);
49         sea((s<<1)+1);
50     }
51 }
52 void dfs1(int u)
53 {
54     int j=head[u];
55     siz[u]=1;
56     while (j>0)
57     {
58         if (d[e[j]]==0)
59         {
60             d[e[j]]=d[u]+1;
61             f[e[j]]=u;
62             dfs1(e[j]);
63             if (siz[e[j]]>siz[s[u]])
64                 s[u]=e[j];
65             siz[u]+=siz[e[j]];
66         }
67         j=nex[j];
68     }
69 }
70 void dfs2(int u)
71 {
72     int j=head[u];
73     if (s[u]!=0)
74     {
75         w[s[u]]=++tot;
76         top[s[u]]=top[u];
77         dfs2(s[u]);
78     }
79     while (j>0)
80     {
81         if (e[j]!=s[u]&&e[j]!=f[u])
82         {
83             w[e[j]]=++tot;

```

```

84         top[e[j]]=e[j];
85         dfs2(e[j]);
86     }
87     j=nex[j];
88 }
89 }
90 void init()
91 {
92     mem(head);
93     mem(s);mem(a);mem(d);
94     ee=tot=0;
95 }
96 int main()
97 {
98     scanf("%d",& );
99     while ( --)
100     {
101         init();
102         scanf("%d",&n);
103         build(1,1,n-1);
104         for (i=1;i<=n-1;i++)
105         {
106             scanf("%d%d%d",&u[i],&v[i],&c[i]);
107             add(u[i],v[i],c[i]);
108             add(v[i],u[i],c[i]);
109         }
110         root=1;d[root]=1;top[root]=root;
111         dfs1(root);
112         dfs2(root);
113         for (i=1;i<=n-1;i++)
114         {
115             if (d[u[i]]>d[v[i]])
116                 swap(u[i],v[i]);
117             ll=w[v[i]];rr=c[i];
118             add(1);
119         }
120         while (1)
121         {
122             getchar();
123             scanf("%s",opt);
124             if (opt[0]=='D')break;
125             scanf("%d%d",&uu,&vv);
126             if (opt[0]=='Q')
127             {
128                 ma=0;
129                 while (top[uu]!=top[vv])
130                 {
131                     if (d[top[uu]]<d[top[vv]])
132                         swap(uu,vv);
133                     ans=0;ll=w[top[uu]];rr=w[uu];sea(1);
134                     ma=MAX(ma,ans);
135                     uu=f[top[uu]];
136                 }
137                 if (uu!=vv)
138                 {
139                     if (d[uu]<d[vv]) swap(uu,vv);
140                     ans=0;ll=w[s[vv]];rr=w[uu];sea(1);
141                     ma=MAX(ma,ans);

```

```

142         }
143         printf("%d\n",ma);
144     }
145     if (opt[0]=='C')
146     {
147         ll=w[v[uu]];rr=vv;
148         add(1);
149     }
150 }
151 }
152 return 0;
153 }
154 /*
155 14
156 1 2
157 2 5
158 2 6
159 6 11
160 6 12
161 1 3
162 3 7
163 1 4
164 4 8
165 4 9
166 4 10
167 9 13
168 13 14
169 */

```

13、FFT 快速卷积

```

1  #include <cstdio>
2  #include <cstring>
3  #include <iostream>
4  #include <algorithm>
5  #include <cmath>
6  #define PI acos(-1.0)
7  #define int long long
8  #define fi first
9  #define se second
10 #define mp make pair
11 using namespace std;
12 typedef long long ll;
13 const int N=60010;
14 ll a[N*16],b[N*16],n;
15 pair <ll,ll> e[N*16];
16 //复数结构体
17 struct complex
18 {
19     long double r,i;
20     complex(long double r = 0.0,long double i = 0.0)
21     {
22         r = _r; i = _i;

```

```

23     }
24     complex operator +(const complex &b)
25     {
26         return complex(r+b.r,i+b.i);
27     }
28     complex operator -(const complex &b)
29     {
30         return complex(r-b.r,i-b.i);
31     }
32     complex operator *(const complex &b)
33     {
34         return complex(r*b.r-i*b.i,r*b.i+i*b.r);
35     }
36 };
37 void change(complex y[],int len)
38 {
39     int i,j,k;
40     for(i = 1, j = len/2; i < len-1; i++)
41     {
42         if(i < j) swap(y[i],y[j]);
43         //交换互为小标反转的元素, i<j 保证交换一次
44         //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
45         k = len/2;
46         while( j >= k)
47         {
48             j -= k;
49             k /= 2;
50         }
51         if(j < k) j += k;
52     }
53 }
54 /*
55  * 做 FFT
56  * len 必须为 2^k 形式,
57  * on==1 时是 DFT, on==-1 时是 IDFT
58  */
59 void fft(complex y[],int len,int on)
60 {
61     change(y,len);
62     for(int h = 2; h <= len; h <<= 1)
63     {
64         complex wn(cos(-on*2*PI/h),sin(-on*2*PI/h));
65         for(int j = 0; j < len; j+=h)
66         {
67             complex w(1,0);
68             for(int k = j; k < j+h/2; k++)
69             {
70                 complex u = y[k];
71                 complex t = w*y[k+h/2];
72                 y[k] = u+t;
73                 y[k+h/2] = u-t;
74                 w = w*wn;
75             }
76         }
77     }
78 }
79 if(on == -1)
80     for(int i = 0; i < len; i++)

```

```

81         y[i].r /= len;
82     }
83     const int MAXN = N*16;
84     complex x1[MAXN],x2[MAXN];
85     int sum[MAXN];
86     bool cmp(pair <ll,ll> a,pair <ll,ll> b){
87         return (a.fi>b.fi);
88     }
89     main()
90     {
91         int ;
92         scanf("%lld",& );
93         while( --)
94         {
95             scanf("%lld",&n);
96             ll s=0;
97             memset(a,0,sizeof(a));
98             memset(b,0,sizeof(b));
99             for (ll i=0;i<n;i++)
100                 scanf("%lld",&a[i]),s+=a[i]*a[i];
101             for (ll i=0;i<n;i++)
102                 scanf("%lld",&b[i]),s+=b[i]*b[i];
103             for (ll i=0;i<n/2;i++)
104                 swap(b[i],b[n-i-1]); //因为后面 n 变了
105             ll x=1;
106             while (x<n*2) x<<=1;
107             ll nn=n;
108             ll len=n*x;
109             for(int i = 0;i < n;i++)
110                 x1[i] = complex(a[i],0);
111             for(int i = 0;i < n;i++)
112                 x2[i] = complex(b[i],0);
113             //求 DFT
114             fft(x1,len,1);
115             fft(x2,len,1);
116             for(int i = 0;i < len;i++)
117                 x1[i] = x1[i]*x2[i];
118             fft(x1,len,-1);
119             for(int i = 0;i < len;i++){
120                 sum[i] = (long long) (x1[i].r+0.5);
121             }
122             ll ans=0;
123             n=nn;
124             for (ll i=0;i<n;i++)
125                 e[i]=mp(sum[i]+sum[i+n],i);
126             sort(e,e+n,cmp);
127             for (ll i=n;i<n*2;i++)
128                 a[i]=a[i-n];
129             for (ll i=0;i<min(10011,n);i++){
130                 ll t=e[i].se+1,p=0;
131                 for (ll j=0;j<n;j++)
132                     p+=a[j+t]*b[n-1-j];
133                 ans=max(ans,p);
134             }
135             printf("%lld\n",s-ans*2);
136         }
137         return 0;
138     }

```

14、Lucas 定理求组合数

```
1  #include<stdio>
2  #define P 10007
3  long long n,m;
4  long long pow(long long a,long long b)
5  {
6      long long ans;
7      ans=1;
8      while (b>0)
9      {
10         if (b%2)
11             ans=(ans*a)%P;
12         b=b>>1;
13         a=(a*a)%P;
14     }
15     return ans;
16 }
17 long long c(long long n,long long m)
18 {
19     long long cc,i;
20     if (n<m) return 0;
21     cc=1;
22     for (i=n;i>=n-m+1;i--) cc=cc*i%P;
23     for (i=2;i<=m;i++)
24         cc=cc*pow(i,P-2)%P;
25     return cc;
26 }
27 long long lucas(long long n,long long m)
28 {
29     if (m==0) return 1;
30     return c(n%P,m%P)*lucas(n/P,m/P)%P;
31 }
32 int main()
33 {
34     while(scanf("%lld%lld",&n,&m)==2)
35     {
36         printf("%lld\n",lucas(n,m));
37     }
38     return 0;
39 }
```

15、欧拉筛法

```
1  #include<stdio>
2  #include<string>
3  #define M 10000000
4  int pp,i,j,n,phi[M],p[M];
```



```

5  bool b[M];
6  int main()
7  {
8      freopen("1.txt", "w", stdout);
9      memset(b, 0, sizeof(b));
10     scanf("%d", &n);
11     phi[1]=1;
12     for (i=2; i<=n; i++)
13     {
14         if (!b[i])
15         {
16             p[++pp]=i;
17             phi[i]=i-1; //phi
18         }
19         for (j=1; j<=pp; j++)
20         {
21             if (p[j]*i>n) break;
22             b[p[j]*i]=true;
23             if (i%p[j]==0)
24             {
25                 phi[p[j]*i]=phi[i]*p[j]; //phi
26                 break;
27             }
28             else
29                 phi[p[j]*i]=phi[i]*(p[j]-1); //phi
30         }
31     }
32     for (i=1; i<=pp; i++) printf("%d", p[i]);
33     printf("%d", pp);
34     return 0;
35 }

```

16、高斯消元

```

1  #include<cstdio>
2  #include<cstring>
3  #include<algorithm>
4  #include<cmath>
5  #define MAX(a,b) a>b?a:b
6  #define N 1100
7  #define NU 1000000
8  #define mem(a) memset(a,0,sizeof(a))
9  using namespace std;
10 double x[N],ma[N][N],a[N][N],m;
11 int i,j,k,n;
12 int main()
13 {
14     scanf("%d",&n);
15     for (i=1;i<=n+1;i++)
16         for (j=1;j<=n;j++)
17             scanf("%lf",&a[i][j]);
18     for (i=1;i<=n;i++)
19         for (j=1;j<=n;j++)

```

```

20     {
21         ma[i][j]=(a[i+1][j]-a[i][j])*2;
22         ma[i][n+1]+=a[i+1][j]*a[i+1][j]-a[i][j]*a[i][j];
23     }
24
25     for (i=1;i<=n-1;i++)
26     {
27         for (j=i;j<=n;j++)
28             if (fabs(ma[j][i])>fabs(ma[i][i]))
29                 for (k=1;k<=n+1;k++)
30                     swap(ma[i][k],ma[j][k]);
31         if (ma[i][i]==0)continue;
32         for (j=i+1;j<=n;j++)
33         {
34             m=ma[j][i]/ma[i][i];
35             ma[j][i]=0;
36             for (k=i+1;k<=n+1;k++)
37                 ma[j][k]=ma[i][k]*m-ma[j][k];
38         }
39     }
40     x[n]=ma[n][n+1]/ma[n][n];
41     for (i=n-1;i>=1;i--)
42     {
43         m=0;
44         for (j=i+1;j<=n;j++)
45             m+=ma[i][j]*x[j];
46         x[i]=(ma[i][n+1]-m)/ma[i][i];
47     }
48     for (i=1;i<n;i++)
49         printf("%.3f ",x[i]);
50     printf("%.3f\n",x[n]);
51 }

```

17、树状数组

```

1  int lowbit(int x)
2  {
3      return (x&(-x));
4  }
5  int search(int x)
6  {
7      int sea=0;
8      while (x>0)
9      {
10         sea+=c[x];
11         x-=lowbit(x);
12     }
13     return sea;
14 }
15 void change(int x,int a)
16 {
17     while (x<=n)
18     {

```

```

19         c[x]+=a;
20         x+=lowbit(x);
21     }
22 }

```

18、二维树状数组

```

1  #include<stdio>
2  #include<cstring>
3  int c[1100][1100],n,p,i,j, ,x1,x2,y1,y2;
4  char opt[10];
5  int lowbit(int x)
6  {
7      return x&(-x);
8  }
9  int search(int x,int y)
10 {
11     int i=x,j=y,ans=0;
12     while (i>0)
13     {
14         j=y;
15         while (j>0)
16         {
17             ans+=c[i][j];
18             j-=lowbit(j);
19         }
20         i-=lowbit(i);
21     }
22     return ans;
23 }
24 void change(int x,int y,int v)
25 {
26     int i=x,j=y;
27     while (i<=n)
28     {
29         j=y;
30         while (j<=n)
31         {
32             c[i][j]+=v;
33             j+=lowbit(j);
34         }
35         i+=lowbit(i);
36     }
37 }
38 int main()
39 {
40     scanf("%d",& );
41     while ( --)
42     {
43         memset(c,0,sizeof(c));
44         scanf("%d%d",&n,&p);
45         for (i=1;i<=p;i++)
46         {

```

```

47         scanf("%s",opt);
48         if (opt[0]=='Q')
49         {
50             scanf("%d%d",&x1,&y1);
51             printf("%d\n",search(x1,y1)%2);
52         }
53         if (opt[0]=='C')
54         {
55             scanf("%d%d%d%d",&x1,&y1,&x2,&y2);
56             change(x1,y1,1);
57             change(x1,y2+1,1);
58             change(x2+1,y1,1);
59             change(x2+1,y2+1,1);
60         }
61     }
62     if (>0) puts("");
63 }
64 return 0;
65 }

```

19、小根堆

```

1 void up()
2 {
3     int i=n;
4     while (a[i]<a[i>>1] && (i>>1)>0)
5     {
6         swap(a[i],a[i>>1]);
7         i>>=1;
8     }
9 }
10 void down()
11 {
12     int now=1,i;
13     while (now<=n/2)
14     {
15         i=now*2;
16         if (i<n&&a[i]>a[i+1]) i++;
17         if (a[now]>a[i])
18         {
19             swap(a[i],a[now]);
20             now=i;
21         }
22         else
23             break;
24     }
25 }

```

20、圆面积交

```
1 double calc(double x1,double y1,double r1,double x2,double y2,double r2)
2 {
3     double d=sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
4     if(r1>r2)
5     {
6         double temp=r1;
7         r1=r2;
8         r2=temp;
9     }
10    if(r1+r2<=d)
11        return 0;
12    else if(r2-r1>=d)
13        return pi*r1*r1;
14    else
15    {
16        double a1=acos((r1*r1+d*d-r2*r2)/(2.0*r1*d));
17        double a2=acos((r2*r2+d*d-r1*r1)/(2.0*r2*d));
18        return (a1*r1*r1+a2*r2*r2-r1*d*sin(a1));
19    }
20 }
```

21、Contor 展开

```
1 #include <stdio>
2 #include <cstring>
3 using namespace std;
4 const int fac[8] = {1,1,2,6,24,120,720,5040};
5 struct node
6 {
7     int a[8];
8     int n;
9 }u,v;
10 int contor(node &t)
11 {
12     int tmp,num=0;
13     for (int i = 0; i < 8; i++)
14     {
15         tmp = 0;
16         for (int j = i + 1; j < 8; j++)
17         {
18             if (t.a[j] < t.a[i])
19                 tmp++;
20         }
21         num+=tmp*fac[7 - i];
22     }
23     return num;
24 }
25 int main()
26 {
27     int i;
```

```

28     for (i=0;i<8;i++)
29         scanf("%d",&u.a[i]);
30     printf("%d\n",contor(u));
31 }

```

22、Splay tree

```

1  #include<cstdio>
2  #define N 200000
3  #define INF 2000000000
4  int s[N][3],f[N],cnt[N],data[N],root,total;//sonlleft 2right
5  int le[N],m,x,n,i,j,ans;
6  char opt;
7  void rotate(int x,int w)//w:1leftZag 2rightZig
8  {
9      int y;
10     y=f[x];
11     cnt[y]=cnt[y]-cnt[x]+cnt[s[x][w]];
12     cnt[x]=cnt[x]-cnt[s[x][w]]+cnt[y];
13     s[y][3-w]=s[x][w];
14     if (s[x][w]!=0)
15         f[s[x][w]]=y;
16     f[x]=f[y];
17     if (f[y]!=0)
18         if (y==s[f[y]][1])
19             s[f[y]][1]=x;
20         else
21             s[f[y]][2]=x;
22     f[y]=x;
23     s[x][w]=y;
24 }
25 void splay(int x,int t)
26 {
27     int y;
28     while (f[x]!=t)
29     {
30         y=f[x];
31         if (f[y]==t)
32             if (x==s[y][1])
33                 rotate(x,2);
34             else
35                 rotate(x,1);
36         else
37             if (y==s[f[y]][1])
38                 if (x==s[y][1])
39                     {
40                         rotate(y,2);
41                         rotate(x,2);
42                     }
43             else
44                 {
45                     rotate(x,1);
46                     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;

```

```

105         x=s[x][2];
106     }
107 }
108 total++;
109 data[total]=w;
110 f[total]=x;
111 cnt[total]=1;
112 if (w<=data[x])
113     s[x][1]=total;
114 else
115     s[x][2]=total;
116 splay(total,0);
117 }
118
119 int extract(int x,int w)//w=1min w=2max
120 {
121     int i=x;
122     if (w==1)
123         i=sea(x,-INF);
124     else
125         i=sea(x,INF);
126     splay(i,0);
127     return i;
128 }
129 void clear(int x)
130 {
131     cnt[x]=s[x][1]=s[x][2]=f[x]=data[x]=0;
132 }
133 void del(int w)
134 {
135     int i;
136     splay(w,0);
137     i=w;
138     if (s[i][1]==0)
139     {
140         root=s[i][2];
141         f[root]=0;
142     }
143     if (s[i][2]==0)
144     {
145         root=s[i][1];
146         f[root]=0;
147     }
148     if (s[i][2]!=0 && s[i][1]!=0)
149     {
150         f[s[i][1]]=0;
151         root=extract(s[i][1],2);
152         s[root][2]=s[i][2];
153         cnt[root]+=cnt[s[i][2]];
154         f[s[root][2]]=root;
155     }
156     clear(i);
157 }
158 int find(int k,int w)//w=1 Kthsmall;w=2 Kthbig
159 {
160     int i,t;
161     i=root;t=k;
162     while (t!=cnt[s[i][w]]+1)

```



```

163     {
164         if (t>cnt[s[i][w]]+1)
165         {
166             t-=cnt[s[i][w]]+1;
167             i=s[i][3-w];
168         }
169         else
170             i=s[i][w];
171     }
172     splay(i,0);
173     return i;
174 }
175 int main()
176 {
177     scanf("%d%d",&n,&m);
178     add(10000000);
179     for (i=1;i<=n;i++)
180     {
181         getchar();
182         scanf("%c %d",&opt,&x);
183         if (opt=='I')
184             if (x>=m)
185                 add(x);
186         if (opt=='A')
187         {
188             for (j=1;j<=total;j++)
189                 data[j]+=x;
190         }
191         if (opt=='F')
192         {
193             x++;
194             if (total-ans<x)
195                 printf("-1\n");
196             else
197                 printf("%d\n",data[find(x,2)]);
198         }
199         if (opt=='S')
200         {
201             for (j=1;j<=total;j++)
202                 data[j]-=x;
203             for (j=1;j<=total;j++)
204                 if (data[j]<m && le[j]==0)
205                 {
206                     le[j]=1;
207                     ans++;
208                     del(j);
209                 }
210         }
211     }
212     printf("%d\n",ans);
213     return 0;
214 }

```

23、快速排序

```
1 void sort(int l,int r)
2 {
3     int i,j,x;
4     i=l;j=r;x=a[(l+r)/2];
5     while (i<=j)
6     {
7         while (a[i]<x) i++;
8         while (a[j]>x) j--;
9         if (i<=j)
10        {
11            swap(a[i],a[j]);
12            i++;j--;
13        }
14    }
15    if (l<j) sort(l,j);
16    if (i<r) sort(i,r);
17 }
```

24、头文件

```
1 #include <map>
2 #include <set>
3 #include <stack>
4 #include <queue>
5 #include <cmath>
6 #include <ctime>
7 #include <vector>
8 #include <cstdio>
9 #include <cctype>
10 #include <cstring>
11 #include <cstdlib>
12 #include <iostream>
13 #include <algorithm>
14 #define ll long long
15 #define ull unsigned long long
16 #define all(x) (x).begin(), (x).end()
17 #define ayacin ios::sync_with_stdio(false);
18 #define yukari 1000000000000000LL;
19 #define mod 1000000007
20 #define M_PI12 3.141592653
21 #define Parsee long double
22 #define reimu iterator
23 #define x first
24 #define y second
25 #define mokou __builtin_popcount
26 #define __sum accumulate
27 #define rep(i,a,b) for(i=a;i<=b;i++)
28 #define lson l , m , rt << 1
29 #define rson m + 1 , r , rt << 1 | 1
30 #define eps 1e-10
```

```

31 #define zero(a) fabs(a)<eps
32
33 struct qwe
34 {
35     int x,c;
36     bool operator < (const qwe &t) const
37     {
38         return x>t.x;
39     }
40 };
41 struct wer
42 {
43     int x,y,c;
44 }e[5000001];
45 priority_queue<qwe>que;

```

25、bsgs

```

1  #include<cstdio>
2  #include<cmath>
3  #include<iostream>
4  #include<cstring>
5  #include<algorithm>
6  #include<cmath>
7  #include<map>
8  #include<set>
9  using namespace std;
10 #define ll    int64
11 #define uint  unsigned int
12 #define RE    freopen("1.in","r",stdin)
13
14 class hash {
15 public:
16     hash() {
17         memset(a,0xff,sizeof(a));
18     }
19     int locate(ll x) {
20         ll l=x%MOD;
21         while(a[l]!=x&&a[l]!=-1) l=(l+1)%MOD;
22         return l;
23     }
24     void insert(ll x,ll va) {
25         ll l=locate(x);
26         if(a[l]==-1) {
27             a[l]=x;
28             v[l]=va;
29         }
30     }
31     ll get(ll x) {
32         ll l=locate(x);
33         return a[l]==x?v[l]:-1;
34     }

```

```

35     void clear() {
36         memset(a,0xff,sizeof(a));
37     }
38 private:
39     static const ll MOD=100007;
40     ll a[MOD+100],v[MOD+100];
41 } S;
42
43 ll exgcd(ll a,ll b,ll &x,ll &y) {
44     ll t,ret;
45     if (!b) {
46         x=1,y=0;
47         return a;
48     }
49     ret=exgcd(b,a%b,x,y);
50     t=x,x=y,y=t-a/b*y;
51     return ret;
52 }
53
54 int main() {///A^x=B(modC)    A^j=B*A^(-m*i)(mod C)
55     ll C,A,B;
56     ll m,i,t,D,ans,x,y;
57     while(scanf("%lld%lld%lld",&C,&A,&B)!=EOF) {
58         S.clear();
59         m=ceil(sqrt((double)C));
60         t=1;
61         for(i=0; i<m; i++) {
62             S.insert(t,i);
63             t=t*A%C;
64         }
65         D=1;///此时 t=A^m
66         ans=-1;
67         for(i=0; i<m; i++) {
68             exgcd(D,C,x,y);///exgcd 求逆元, 得到 x=D^(-i*m)
69             x=(x*B)%C+C%C;///B*x=B*D^(-i*m)
70             y=S.get(x);
71             //printf("%lld,%lld\n",x,y);
72             if(y!=-1) {
73                 ans=i*m+y;
74                 break;
75             }
76             D=(D*t)%C;///D=t^i, (t=A^m)
77         }
78         if(ans==-1) printf("no solution\n");
79         else printf("%lld\n",ans);
80     }
81     return 0;
82 }

```

26、莫队算法

```

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

```

```

3  #define se second
4  #define mp make pair
5  #define pb push back
6  using namespace std;
7  const int N = 50010;
8  struct Query
9  {
10     int L,R,id;
11 } node[N];
12 int ans[N],a[N],b[N],n,m,k,unit,i;
13 pair<int,int> t[100010];
14 bool cmp(Query a,Query b)
15 {
16     if(a.L/unit != b.L/unit) return a.L/unit < b.L/unit;
17     else return a.R < b.R;
18 }
19 void solve()
20 {
21     int temp = 0;
22     memset(t,0,sizeof(t));
23     int L = 1;
24     int R = 0;
25     for(int i = 1; i <=m; i++)
26     {
27         while(R < node[i].R)
28         {
29             R++;
30             if (t[a[R]].fi<t[a[R]].se) temp++;
31             t[a[R]].fi++;
32             if (t[b[R]].se<t[b[R]].fi) temp++;
33             t[b[R]].se++;
34         }
35         while(R > node[i].R)
36         {
37             if (t[a[R]].fi<=t[a[R]].se) temp--;
38             t[a[R]].fi--;
39             if (t[b[R]].se<=t[b[R]].fi) temp--;
40             t[b[R]].se--;
41             R--;
42         }
43         while(L < node[i].L)
44         {
45             if (t[a[L]].fi<=t[a[L]].se) temp--;
46             t[a[L]].fi--;
47             if (t[b[L]].se<=t[b[L]].fi) temp--;
48             t[b[L]].se--;
49             L++;
50         }
51         while(L > node[i].L)
52         {
53             L--;
54             if (t[a[L]].fi<t[a[L]].se) temp++;
55             t[a[L]].fi++;
56             if (t[b[L]].se<t[b[L]].fi) temp++;
57             t[b[L]].se++;
58         }
59         ans[node[i].id]=temp;
60     }

```

```

61 }
62 int main()
63 {
64     scanf("%d%d%d",&n,&m,&k);
65     for (i=1; i<=n; i++) scanf("%d",&a[i]);
66     for (i=1; i<=n; i++) scanf("%d",&b[i]);
67     for(i = 1; i<=m; i++)
68     {
69         node[i].id = i;
70         scanf("%d%d",&node[i].L,&node[i].R);
71         node[i].L++,node[i].R++;
72     }
73     unit = (int)sqrt(n);
74     sort(node+1,node+m+1,cmp);
75     solve();
76     for (i=1; i<=m; i++)
77         printf("%d\n",ans[i]);
78     return 0;
79 }
80 /*
81 3 4 2
82 0 0 1
83 0 0 0
84 0 0
85 2 2
86 0 1
87 1 2
88
89 7 10 3
90 0 1 2 1 1 0 2
91 1 0 0 1 0 2 1
92 */

```

27、treap

```

1 struct Node{
2     Node *s[2];
3     int v, w, pos, mk, cnt;
4     //ll r[5], p;
5     //ll getv(int i = 0){ return r[(p % 5 + i + 5) % 5]; }
6     Node *link(int w, Node *p){ s[w] = p; return this; }
7     void init(){ memset(this, 0, sizeof(*this)); w = rand(); cnt = 1; }
8     void push();
9     void update();
10 };
11 struct HNIUNO{
12     static const int size = 1e6 + 5;
13     Node spc[size];
14     int p;
15     void reset(){ p = 0; }
16     Node *nloc(){ spc[p].init(); return spc + (p++); }
17 }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

```
1  typedef long long LL;
2  const int maxn=200000;
3  const LL P=1004535809; //P=C*2^k+1, P 是质数
4  const LL g=3; //P 的原根
5  int N,na,nb;
6  int a[maxn*2],b[maxn*2],W[2][maxn*2],rev[maxn*2];
7
8  LL Pow(LL a,int b)
9  {
10     LL c=1;
11     for (;b>=1;a=a*a%P) if (b&1) c=c*a%P;
12     return c;
13 }
14 void NTT(int*a,int f)
15 {
16     for (int i=0;i<N;i++) if (i<rev[i]) swap(a[i],a[rev[i]]);
17     for (int i=1;i<N;i<=1)
18         for (int j=0,t=N/(i<=1);j<N;j+=i<=1)
19             for (int k=0,l=0,x,y;k<i;k++,l+=t)
20                 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;
21     if (f) for (int i=0,x=Pow(N,P-2);i<N;i++) a[i]=(LL)a[i]*x%P;
22 }
23 void change()
24 {
25     for (int i=0;i<N;i++)
26     {
27         int x=i,y=0;
28         for (int k=1; k<N; x>=1,k<=1) (y<=1)|=x&1;
29         rev[i]=y;
30     }
31 }
32 int main()
33 {
34     for (N=1; N<na||N<nb; N<=1); N<=1;
35     for (int i=na;i<N;i++)
36         a[i]=b[i]=0;
37     W[0][0]=W[1][0]=1;
38     for (int i=1,x=Pow(g,(P-1)/N),y=Pow(x,P-2); i<N; i++)
39         W[0][i]=(LL)x*W[0][i-1]%P,W[1][i]=(LL)y*W[1][i-1]%P;
40     change();
41     NTT(a,0),NTT(b,0);
42     for (int i=0;i<N;i++) a[i]=(LL)a[i]*b[i]%P;
43     NTT(a,1);
44 }
```

29、Java 大数 a+b

```
1  import java.math.BigInteger;
2  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 }

```