

Contents

- 1 费用流
- 2 hdu2665(可持久线段树)
- 3 hdu2890(suffixArray)
- 4 hdu3691(dinic)
- 5 hdu4992
- 6 hdu5081(cdq)
- 7 hdu5127(px+qy)
- 8 hdu5130(圆与多边形交)
- 9 hdu5384(ac)
- 10 poj2417(bsgs)
- 11 tarjan

1 费用流

```
1 1 const int inf=~0u>>1;
2 2 int n,m;
3 3 int S,T;
4 4 int ans;
5 5 struct node{
6 6     int to,f,v,next;
7 7 } w[1000001];
8 8 int h[1001],cnt=1;
9 9 void add(int x,int y,int f,int v) {
10 10     w[++cnt]=(node){y,f,v,h[x]};
11 11     h[x]=cnt;
12 12     w[++cnt]=(node){x,0,-v,h[y]};
13 13     h[y]=cnt;
14 14 }
15 15 queue<int> q;
16 16 int d[1001];
17 17 bool vst[1001];
18 18 int fa[1001];
19 19 bool Spfa() {
20 20     int i,j,v;
21 21     while (!q.empty()) q.pop();
22 22     memset(vst,0,sizeof vst);
23 23     for (i=1; i<=T; i++) d[i]=-inf;
24 24     memset(fa,0,sizeof fa);
25 25     d[S]=0;
26 26     vst[S]=1;
27 27     q.push(S);
28 28     while (!q.empty()) {
29 29         i=q.front();
30 30         q.pop();
31 31         vst[i]=0;
32 32         for (v=h[i]; v; v=w[v].next) {
33 33             j=w[v].to;
34 34             if (w[v].f&& d[i]+w[v].v>d[j]) {
35 35                 d[j]=d[i]+w[v].v;
36 36                 fa[j]=v;
37 37                 if (!vst[j]) {
38 38                     vst[j]=1;
39 39                     q.push(j);
40 40             }
41 41         }
42 42     }
43 43 }
44 44 return d[T]!=-inf;
45 45 }
46 46 void Adjust() {
```

```

47     int i,v,f=inf;
48     i=T;
49     while (i!=S) {
50         v=fa[i];
51         f=min(f,w[v].f);
52         i=w[v^1].to;
53     }
54     ans+=f*d[T];
55     i=T;
56     while (i!=S) {
57         v=fa[i];
58         w[v].f-=f;
59         w[v^1].f+=f;
60         i=w[v^1].to;
61     }
62 }
63 int main() {
64     while (Spfa()) Adjust();
65 }

```

2 hdu2665(可持久线段树)

```

1 #include<iostream>
2 #include<cstdio>
3 #include<algorithm>
4 #define mp make_pair
5 #define st first
6 #define nd second
7 using namespace std;
8 const int inf=~0u>>1;
9 int n,m;
10 int ans;
11 int a[100001];
12 pair<int,int> b[100001];
13 int f[100001];
14 struct ST{
15     int ls,rs,v;
16 } T[2000001];
17 int tot;
18 int h[100001];
19 void build(int u,int l,int r) {
20     int mid;
21     if (l==r) return;
22     mid=l+r>>1;
23     T[u].ls=++tot;
24     T[u].rs=++tot;
25     T[u].v=0;

```

```

26     build(T[u].ls,l,mid);
27     build(T[u].rs,mid+1,r);
28 }
29 int add(int u,int l,int r,int k) {
30     int mid;
31     T[++tot]=T[u];
32     u=tot;
33     T[u].v++;
34     if (l==r) return u;
35     mid=l+r>>1;
36     if (k<=mid)
37         T[u].ls=add(T[u].ls,l,mid,k);
38     else T[u].rs=add(T[u].rs,mid+1,r,k);
39     return u;
40 }
41 int ask(int u,int v,int l,int r,int k) {
42     int mid,t;
43     if (l==r) return l;
44     mid=l+r>>1;
45     t=T[T[v].ls].v-T[T[u].ls].v;
46     if (t>=k) return ask(T[u].ls,T[v].ls,l,mid,k);
47     else return ask(T[u].rs,T[v].rs,mid+1,r,k-t);
48 }
49 int main() {
50     int __,i,j,l,r,v;
51     scanf("%d",&__);
52     while (__--) {
53         scanf("%d%d",&n,&m);
54         for (i=1; i<=n; i++) {
55             scanf("%d",&v);
56             b[i]=mp(v,i);
57         }
58         sort(b+1,b+n+1);
59         b[0]=mp(inf,inf);
60         for (i=1,j=0; i<=n; i++) {
61             if (b[i]!=b[i-1]) {
62                 j++;
63                 f[j]=b[i].st;
64             }
65             a[b[i].nd]=j;
66         }
67         tot=1;
68         build(1,1,n);
69         h[0]=1;
70         for (i=1; i<=n; i++)
71             h[i]=add(h[i-1],1,n,a[i]);
72         for (i=1; i<=m; i++) {
73             scanf("%d%d%d",&l,&r,&v);

```

```

74         ans=ask(h[l-1],h[r],1,n,v);
75         ans=f[ans];
76         printf("%d\n",ans);
77     }
78 }
79 return 0;
80 }

```

3 hdu2890(suffixArray)

```

1 #include<iostream>
2 #include<cstdio>
3 #include<cstring>
4 #include<algorithm>
5 #include<vector>
6 #include<map>
7 #include<cstdlib>
8 #include<ctime>
9 #include<stdlib>
10 #define pb push_back
11 #define mp make_pair
12 #define st first
13 #define nd second
14 using namespace std;
15 typedef vector<int>::iterator vit;
16 typedef map<int,int>::iterator mit;
17 const int inf=~0u>>2;
18 const int N=200000;
19 int n,m;
20 time_t temp;
21 struct SA{
22     int a[N+1];
23     int c[N+1];
24     int len;
25     int h[N+1];
26     pair<int,int> b[N+1];
27     int hs[N+1];
28     int rk[N+1];
29     int tab[N+1];
30     int vw[N+1];
31     void sa() {
32         int i,j,k,x,y;
33         memcpy(c,a,(len+1)*sizeof(int));
34         b[0]=mp(inf,inf);
35         for (i=1; i<=len; i++)
36             b[i]=mp(a[i],i);
37         sort(b+1,b+len+1);

```

```

38         for (i=1,j=0; i<=len; i++) {
39             if (b[i].st!=b[i-1].st) j++;
40             a[b[i].nd]=j;
41         }
42         memset(hs,0,(len+1)*sizeof(int));
43         for (i=1; i<=len; i++)
44             hs[a[i]]++;
45         for (i=1; i<=len; i++)
46             hs[i]+=hs[i-1];
47         for (i=len; i; i--)
48             tab[hs[a[i]]--]=i;
49         for (i=1; i<=len; i<=1) {
50             for (j=len-i+1,k=0; j<=len; j++)
51                 rk[++k]=j;
52             for (j=1; j<=len; j++)
53                 if (tab[j]>i) rk[++k]=tab[j]-i;
54             memset(hs,0,(len+1)*sizeof(int));
55             for (j=1; j<=len; j++)
56                 hs[a[rk[j]]]++;
57             for (j=1; j<=len; j++)
58                 hs[j]+=hs[j-1];
59             for (j=len; j; j--)
60                 tab[hs[a[rk[j]]]--]=rk[j];
61             for (j=1,k=0; j<=len; j++) {
62                 x=tab[j];
63                 y=tab[j-1];
64                 if (a[x]!=a[y] || a[x+i]!=a[y+i]) k++;
65                 vw[tab[j]]=k;
66             }
67             for (j=1; j<=len; j++) a[j]=vw[j];
68         }
69         for (i=1; i<=len; i++) {
70             if (a[i]==1) h[i]=0;
71             else {
72                 j=tab[a[i]-1];
73                 k=max(0,h[i-1]-1);
74                 while (i+k<=len && j+k<=len && c[i+k]==c[j+k]) k++;
75                 h[i]=k;
76             }
77         }
78     }
79 } S;
80 int a[N+1];
81 int h[N+1];
82 int q[N+1],top;
83 int jud(int mid) {
84     int i,j,k,t,ret=0;
85     for (i=1; i<n; i++) {

```

```

86     j=i;
87     top=0;
88     q[++top]=a[j];
89     while (j<n&&h[a[j+1]]>=mid) {
90         j++;
91         q[++top]=a[j];
92     }
93     if (j-i+1>=m) {
94         sort(q+1,q+top+1);
95         ret=0;
96         t=-inf;
97         for (i=1; i<=top; i++) {
98             if (q[i]-t>=mid) ret++,t=q[i];
99             if (ret+top-i<m) break;
100         }
101         if (ret>=m) return q[1];
102     }
103     i=j;
104     if (n-i<m) break;
105 }
106 return 0;
107 }
108 void binary() {
109     int l=1,r=n,mid;
110     while (l<r) {
111         mid=l+r+1>>1;
112         if (jud(mid)) l=mid;
113         else r=mid-1;
114     }
115     int i,j;
116     j=jud(l);
117     if (j) {
118         printf("%d\n",l);
119         for (i=j; i<=j+l-1; i++)
120             printf("%d\n",S.c[i]);
121     }
122     else printf("0\n");
123 }
124 int main() {
125     int __,i,x;
126     scanf("%d",&__);
127     while (__--) {
128         scanf("%d%d",&n,&m);
129
130         memset(S.a,0,(2*n+1)*sizeof(int));
131         S.len=n;
132         for (i=1; i<=n; i++) {
133             scanf("%d",&x);

```

```

134         S.a[i]=x;
135     }
136     S.sa();
137     memcpy(a,S.tab,(n+1)*sizeof(int));
138     memcpy(h,S.h,(n+1)*sizeof(int));
139     binary();
140     if (__) printf("\n");
141 }
142 return 0;
143 }

```

4 hdu3691(dinic)

```

1 #include<iostream>
2 #include<cstdio>
3 #include<cstring>
4 #include<queue>
5 using namespace std;
6 const int inf=~0u>>1;
7 int n,m;
8 int S,T;
9 int ans;
10 struct node{
11     int to,f,v,next;
12 } w[100010];
13 int h[301],cnt=1;
14 void add(int x,int y,int z) {
15     w[++cnt]=(node){y,z,z,h[x]};
16     h[x]=cnt;
17     w[++cnt]=(node){x,z,z,h[y]};
18     h[y]=cnt;
19 }
20 queue<int> q;
21 int d[301];
22 bool bfs(int S) {
23     int i,j,v;
24     memset(d,0,sizeof d);
25     d[S]=1;
26     q.push(S);
27     while (!q.empty()) {
28         i=q.front();
29         q.pop();
30         for (v=h[i]; v; v=w[v].next) {
31             j=w[v].to;
32             if (w[v].f&&!d[j]) {
33                 d[j]=d[i]+1;
34                 q.push(j);

```

```

35     }
36 }
37 }
38 return d[T];
39 }
40 int dfs(int u,int f) {
41     int j,v,t,ret=0;
42     if (u==T) return f;
43     for (v=h[u]; v; v=w[v].next) {
44         j=w[v].to;
45         if (w[v].f&&d[j]==d[u]+1) {
46             t=dfs(j,min(f,w[v].f));
47             w[v].f-=t,w[v^1].f+=t;
48             f-=t,ret+=t;
49             if (!f) return ret;
50         }
51     }
52     return ret;
53 }
54 int cal() {
55     int i,j,v,ret=0;
56     for (v=h[T]; v; v=w[v].next) {
57         j=w[v].to;
58         ret+=w[v].f-w[v].v;
59     }
60     return ret;
61 }
62 int main() {
63     int i,p,x,y,z;
64     while (scanf("%d%d%d",&n,&m,&S)!=EOF) {
65         if (n==0&&m==0&&S==0) break;
66         memset(h,0,sizeof h);
67         cnt=1;
68         for (i=1; i<=m; i++) {
69             scanf("%d%d%d",&x,&y,&z);
70             add(x,y,z);
71         }
72         ans=inf;
73         p=S;
74         for (i=1; i<=n; i++)
75             if (i!=S) {
76                 T=i;
77                 while (bfs(p)) dfs(p,inf);
78                 while (bfs(S)) dfs(S,inf);
79                 ans=min(ans,cal());
80                 p=i;
81             }
82     printf("%d\n",ans);

```

```

83     }
84     return 0;
85 }

```

5 hdu4992

```

1 #include<iostream>
2 #include<cstdio>
3 #include<cstring>
4 using namespace std;
5 typedef long long ll;
6 int maxx=1000000;
7 int n,m;
8 int p[100001];
9 bool vst[1000001];
10 int fst[1000001];
11 int e[1000001];
12 void Euler() {
13     int i,j,k;
14     for (i=2; i<=maxx; i++) {
15         if (!vst[i]) {
16             p[++p[0]] = i;
17             fst[i]=p[0];
18             e[i] = i-1;
19         }
20         for (j=1; j<=p[0] && i*p[j]<=maxx; j++) {
21             k=i*p[j];
22             vst[k]=1;
23             fst[k]=j;
24             if (i%p[j]==0) {
25                 e[k] = e[i] * p[j];
26                 break;
27             }
28             else {
29                 e[k] = e[i] * (p[j] - 1);
30             }
31         }
32     }
33     //for (i=1; i<=100; i++) cout<<p[i]<<" ";
34     //cout<<endl;
35 }
36 int b[101];
37 void devide(int u) {
38     int i,j;
39     b[0]=0;
40     while (u>1) {
41         i=fst[u];

```

```

42     j=p[i];
43     b[++b[0]]=j;
44     while (u%j==0) u/=j;
45 }
46 //for (i=1; i<=b[0]; i++) cout<<b[i]<<" ";
47 //cout<<endl;
48 }
49 int pow(int a,int b) {
50     int ret=1;
51     while (b) {
52         if (b&1) ret=(ll)ret*a%n;
53         a=(ll)a*a%n;
54         b>>=1;
55     }
56     return ret;
57 }
58 bool jud(int u) {
59     int i;
60     if (pow(u,m)==0) return 0;
61     for (i=1; i<=b[0]; i++)
62         if (pow(u,m/b[i])==1) return 0;
63     return 1;
64 }
65 int main() {
66     int i;
67     Euler();
68     while (scanf("%d",&n)!=EOF) {
69         m = e[n];
70         //cout<<m<<endl;
71         //continue;
72         devide(m);
73         for (i=2; i<n; i++)
74             if (jud(i)) printf("%d ",i);
75         printf("%d\n");
76     }
77     return 0;
78 }

```

6 hdu5081(cdq)

```

1 #include<iostream>
2 #include<cstdio>
3 #include<cstring>
4 #include<algorithm>
5 #include<cmath>
6 #include<queue>
7 #include<map>

```

```

8 #define pb push_back
9 #define mp make_pair
10 #define st first
11 #define nd second
12 using namespace std;
13 typedef long long ll;
14 typedef vector<int>::iterator vit;
15 typedef vector<pair<int,int> >::iterator pit;
16 int n,m;
17 struct Trie{
18     int ch[26];
19     int fail;
20     void init() {
21         memset(ch,0,sizeof ch);
22         fail=0;
23     }
24 } T[100001];
25 vector<int> adj[100001];
26 queue<int> q;
27 void ac() {
28     int i,j,k,fail;
29     while (!q.empty()) q.pop();
30     T[1].fail=0;
31     q.push(1);
32     while (!q.empty()) {
33         i=q.front();
34         q.pop();
35         fail=T[i].fail;
36         for(j=0; j<26; j++) {
37             if (T[i].ch[j]) {
38                 k=T[i].ch[j];
39                 T[k].fail=T[fail].ch[j];
40                 if (!T[k].fail) T[k].fail=1;
41                 q.push(k);
42             }
43             else {
44                 T[i].ch[j]=T[fail].ch[j];
45             }
46         }
47     }
48 }
49 int tot;
50 int f[100001];
51 int fr[100001];
52 vector<int> g[100001];
53 int d[100001];
54 int fa[100001][21];
55 void dfs(int u,int v) {

```

```

56 int j;
57 f[u]=v;
58 for (int i=0; i<16; i++)
59     if (fa[v][i])
60         fa[v][i+1]=fa[fa[v][i]][i];
61     else break;
62 for (vit it=adj[u].begin(); it!=adj[u].end(); it++) {
63     j=*it;
64     tot++;
65     g[v].pb(tot);
66     d[tot]=d[v]+1;
67     fa[tot][0]=v;
68     dfs(j,tot);
69 }
70 fr[v]=tot;
71 }
72 int lca(int u,int v) {
73     int i;
74     if (u==v) return u;
75     if (d[u]>d[v]) swap(u,v);
76     for (i=16; i>=0; i--)
77         if (d[fa[v][i]]>=d[u]) v=fa[v][i];
78     if (u==v) return u;
79     for (i=16; i>=0; i--)
80         if (fa[u][i]!=fa[v][i]) u=fa[u][i],v=fa[v][i];
81     return fa[u][0];
82 }
83 int a[100001];
84 vector<int> b[100001];
85 int c[100001];
86 void ready(int u,int l) {
87     int i,j,k;
88     sort(c+1,c+l+1);
89     if (a[u]==1) {
90         j=0;
91         for (i=1; i<=l; i++)
92             if (c[i]>fr[j]) {
93                 b[u].pb(c[i]);
94                 j=c[i];
95             }
96     }
97     else {
98         j=c[1];
99         b[u].pb(j);
100         for (i=2; i<=l; i++) {
101             k=lca(j,c[i]);
102             b[u].pb(k);
103             b[u].pb(c[i]);

```

```

104         j=c[i];
105     }
106 }
107 }
108 vector<pair<int,int> > vs[400001];
109 vector<int> vt[400001];
110 int qs[200001];
111 void build(int u,int l,int r) {
112     int mid;
113     if (l==r) {
114         vs[u].clear();
115         vt[u].clear();
116         if (a[l]==1) {
117             for (vit it=b[l].begin(); it!=b[l].end(); it++) {
118                 vs[u].pb(mp(*it,1));
119             }
120         }
121         else {
122             int i,j;
123             j=0;
124             for (vit it=b[l].begin(); it!=b[l].end(); it++) {
125                 qs[++j]=*it;
126             }
127             sort(qs+1,qs+j+1);
128             for (i=1; i<=j; i++)
129                 if (qs[i]!=qs[i-1]) vt[u].pb(qs[i]);
130         }
131         return;
132     }
133     mid=l+r>>1;
134     build(u<<1,l,mid);
135     build(u<<1|1,mid+1,r);
136     int i,j,l1,l2,lt,rt;
137
138     lt=u<<1;
139     rt=u<<1|1;
140
141     vs[u].clear();
142     l1=vs[lt].size();
143     l2=vs[rt].size();
144     i=0,j=0;
145     while (i<l1&&j<l2) {
146         int t;
147         t=vs[lt][i].st-vs[rt][j].st;
148         if (t<0) vs[u].pb(vs[lt][i++]);
149         if (t>0) vs[u].pb(vs[rt][j++]);
150         if (t==0) {
151             vs[u].pb(mp(vs[lt][i].st,vs[lt][i].nd+vs[rt][j].nd));

```

```

152         i++;
153         j++;
154     }
155 }
156 while (i<l1) vs[u].pb(vs[l1][i++]);
157 while (j<l2) vs[u].pb(vs[r1][j++]);
158
159 vt[u].clear();
160 l1=vt[l1].size();
161 l2=vt[r1].size();
162 i=0,j=0;
163 while (i<l1&&j<l2) {
164     int t;
165     t=vt[l1][i]-vt[r1][j];
166     if (t<0) vt[u].pb(vt[l1][i++]);
167     if (t>0) vt[u].pb(vt[r1][j++]);
168     if (t==0) {
169         vt[u].pb(vt[l1][i]);
170         i++;
171         j++;
172     }
173 }
174 while (i<l1) vt[u].pb(vt[l1][i++]);
175 while (j<l2) vt[u].pb(vt[r1][j++]);
176
177 }
178 struct sta{
179     int u,v;
180     ll z;
181     sta() {}
182     sta(int u,int v,ll z):u(u),v(v),z(z) {}
183 } stk[100001];
184 int top;
185 ll sum[100001];
186 ll ans[100001];
187 pair<int,int> s[100001];
188 void cdq(int u,int l,int r) {
189     int mid;
190     if (l==r) return;
191     mid=l+r>>1;
192     cdq(u<<1,l,mid);
193
194     if (!vs[u<<1].empty()&&!vt[u<<1|1].empty()) {
195
196         int i,j,k,l1,l2,lt,rt;
197         lt=u<<1;
198         rt=u<<1|1;
199         l1=vs[lt].size();

```

```

200         l2=vt[rt].size();
201         i=0,j=0,k=0;
202         while (i<l1&&j<l2) {
203             int t;
204             t=vs[lt][i]-vt[rt][j];
205             if (t<0) s[++k]=vs[lt][i++];
206             if (t>0) s[++k]=mp(vt[rt][j++],0);
207             if (t==0) {
208                 s[++k]=vs[lt][i];
209                 i++;
210                 j++;
211             }
212         }
213         while (i<l1) s[++k]=vs[lt][i++];
214         while (j<l2) s[++k]=mp(vt[rt][j++],0);
215
216         top=0;
217         for (i=1; i<=k; i++) {
218             while (top&&s[i].st>fr[stk[top].u]) top--;
219             ll t;
220             t=(ll)(d[s[i].st]-d[stk[top].u])*stk[top].v+s[i].nd+stk[top].z;
221             top++;
222             stk[top]=sta(s[i].st,stk[top-1].v+s[i].nd,t);
223             sum[s[i].st]=t;
224         }
225
226         for (i=mid+1; i<=r; i++)
227             if (a[i]==2) {
228                 int k;
229                 k=0;
230                 for (vit it=b[i].begin(); it!=b[i].end(); it++) {
231                     k++;
232                     if (k) ans[i]+=sum[*it];
233                     else ans[i]-=sum[*it];
234                 }
235             }
236
237     }
238
239     cdq(u<<1|1,mid+1,r);
240 }
241 int main() {
242     int __,i,j,k,x;
243     char ch;
244     scanf("%d",&__);
245     while (__) {
246         scanf("%d",&n);
247         for (i=1; i<=n; i++)

```



```

248     T[i].init();
249     for (i=2; i<=n; i++) {
250         scanf("%d %c",&x,&ch);
251         T[x].ch[ch-'a']=i;
252     }
253     ac();
254     for (i=1; i<=n; i++)
255         adj[i].clear();
256     for (i=2; i<=n; i++) {
257         adj[T[i].fail].pb(i);
258     }
259     for (i=1; i<=n; i++) g[i].clear();
260     tot=1;
261     d[1]=1;
262     memset(fa,0,sizeof fa);
263     dfs(1,1);
264
265     scanf("%d",&m);
266     for (i=1; i<=m; i++) b[i].clear();
267     memset(ans,0,sizeof ans);
268     for (i=1; i<=m; i++) {
269         scanf("%d",&a[i]);
270         scanf("%d",&k);
271         for (j=1; j<=k; j++) {
272             scanf("%d",&x);
273             c[j]=f[x];
274         }
275         ready(i,k);
276     }
277     build(1,1,m);
278     cdq(1,1,m);
279     for (i=1; i<=m; i++)
280         if (a[i]==2)
281             printf("%I64d\n",ans[i]);
282 }
283 return 0;
284 }

```

7 hdu5127(px+qy)

```

1 #include<iostream>
2 #include<cstdio>
3 #include<cstring>
4 #include<algorithm>
5 #include<map>
6 #include<vector>
7 #define mp make_pair

```

```

8 #define pb push_back
9 #define st first
10 #define nd second
11 using namespace std;
12 typedef map<pair<int,int>,int>::iterator mit;
13 typedef long long ll;
14 typedef pair<int,int> pa;
15 const ll inf=~0ull>>1;
16 int n;
17 int tot;
18 ll ans[60001];
19 pair<int,int> a[60001];
20 map<pair<int,int>,int> b;
21 pair<pair<int,int>,pair<int,int> > q[60001];
22 int top;
23 vector<pair<int,int> > h[200001];
24 void build(int u,int l,int r) {
25     int mid;
26     mid=l+r>>1;
27     h[u].clear();
28     if (l==r) return;
29     build(u<<1,l,mid);
30     build(u<<1|1,mid+1,r);
31 }
32 void add(int u,int l,int r,int lt,int rt,int x,int y) {
33     int mid;
34     if (l>rt||r<lt) return;
35     if (l>=lt&&r<=rt) {
36         h[u].pb(mp(x,y));
37         return;
38     }
39     mid=l+r>>1;
40     add(u<<1,l,mid,lt,rt,x,y);
41     add(u<<1|1,mid+1,r,lt,rt,x,y);
42 }
43 bool cmp(const pa &a,const pa &b) {
44     return a.nd<b.nd||a.nd==b.nd&&a.st<b.st;
45 }
46 pair<int,int> q1[60001];
47 pair<int,int> q2[60001];
48 int m1,m2;
49 pa operator+(const pa &a,const pa &b) {
50     return mp(a.st+b.st,a.nd+b.nd);
51 }
52 pa operator-(const pa &a,const pa &b) {
53     return mp(a.st-b.st,a.nd-b.nd);
54 }
55 ll operator*(const pa &a,const pa &b) {

```

```

56     return (l1)a.st*b.st+(l1)a.nd*b.nd;
57 }
58 ll operator/(const pa &a,const pa &b) {
59     return (l1)a.st*b.nd-(l1)a.nd*b.st;
60 }
61 ll area(pa p0,pa p1,pa p2) {
62     return (p1-p0)/(p2-p0);
63 }
64 void scan(int u) {
65     int i,l;
66     m1=0;
67     m2=0;
68     l=h[u].size();
69     for (i=0; i<l; i++) {
70         while (m1>1&&area(q1[m1-1],q1[m1],h[u][i])<=0) m1--;
71         q1[++m1]=h[u][i];
72     }
73     for (i=l-1; i>=0; i--) {
74         while (m2>1&&area(q2[m2-1],q2[m2],h[u][i])<=0) m2--;
75         q2[++m2]=h[u][i];
76     }
77 }
78 ll binary(int x,int y,int px,int py) {
79     int l,r,mid;
80     ll ret;
81     l=1;
82     r=m1;
83     while (l<r) {
84         mid=l+r>>1;
85         if ((q1[mid+1]-q1[mid])/mp(x,y)>0) l=mid+1;
86         else r=mid;
87     }
88     ret=mp(px,py)*q1[l];
89     x=-x;
90     y=-y;
91     l=1;
92     r=m2;
93     while (l<r) {
94         mid=l+r>>1;
95         if ((q2[mid+1]-q2[mid])/mp(x,y)>0) l=mid+1;
96         else r=mid;
97     }
98     ret=max(ret,mp(px,py)*q2[l]);
99     return ret;
100 }
101 void cal(int u,int l,int r) {
102     int mid,i,x,y;
103     sort(h[u].begin(),h[u].end(),cmp);

```

```

104     if (!h[u].empty()) {
105         scan(u);
106         for (i=1; i<=r; i++) {
107             x=-a[i].nd,y=a[i].st;
108             if (y<0) x=-x,y=-y;
109             ans[i]=max(ans[i],binary(x,y,a[i].st,a[i].nd));
110         }
111     }
112     if (l==r) return;
113     mid=l+r>>1;
114     cal(u<<1,l,mid);
115     cal(u<<1|1,mid+1,r);
116 }
117 int main() {
118     int i,l,r,t,x,y;
119     while (scanf("%d",&n),n) {
120         tot=0;
121         top=0;
122         b.clear();
123         for (i=1; i<=n; i++) {
124             scanf("%d%d%d",&t,&x,&y);
125             if (!t) a[++tot]=mp(x,y);
126             else if (t==1) {
127                 b[mp(x,y)]=tot+1;
128             }
129             else {
130                 l=b[mp(x,y)];
131                 r=tot;
132                 if (l<=r) q[++top]=mp(mp(x,y),mp(l,r));
133                 b.erase(mp(x,y));
134             }
135         }
136         for (mit it=b.begin(); it!=b.end(); it++) {
137             x=it->st.st;
138             y=it->st.nd;
139             l=it->nd;
140             r=tot;
141             if (l<=r) q[++top]=mp(mp(x,y),mp(l,r));
142         }
143         if (!tot) continue;
144         for (i=1; i<=tot; i++) ans[i]=-inf;
145         build(1,1,tot);
146         for (i=1; i<=top; i++) {
147             x=q[i].st.st;
148             y=q[i].st.nd;
149             l=q[i].nd.st;
150             r=q[i].nd.nd;
151             add(1,1,tot,l,r,x,y);

```

```

152     }
153     cal(1,1,tot);
154     for (i=1; i<=tot; i++)
155         printf("%I64d\n",ans[i]);
156 }
157 return 0;
158 }

```

8 hdu5130(圆与多边形交)

```

1 #include<iostream>
2 #include<cstdio>
3 #include<cstring>
4 #include<cmath>
5 using namespace std;
6 const double PI=acos(-1.0);
7 const double eps=1e-8;
8 int n;
9 double k;
10 double R;
11 double ans;
12 int dcmp(double x) {
13     if (x<-eps) return -1;
14     if (x>eps) return 1;
15     return 0;
16 }
17 struct point{
18     double x,y;
19     point() {}
20     point(double x,double y):x(x),y(y) {}
21     point operator+(const point &a) const{
22         return point(x+a.x,y+a.y);
23     }
24     point operator-(const point &a) const{
25         return point(x-a.x,y-a.y);
26     }
27     double operator*(const point &a) const{
28         return x*a.x+y*a.y;
29     }
30     point operator*(const double &a) const{
31         return point(x*a,y*a);
32     }
33     double operator/(const point &a) const{
34         return x*a.y-y*a.x;
35     }
36     point operator/(const double &a) const{
37         return point(x/a,y/a);

```

```

38     }
39     bool operator==(const point &a) const{
40         return dcmp(x-a.x)==0&&dcmp(y-a.y)==0;
41     }
42     double abs() {
43         return sqrt(x*x+y*y);
44     }
45 } p[1001];
46 point O=point(0,0);
47 point crosspt(point &a,point &b,point &c,point &d) {
48     double s1,s2;
49     s1=(c-a)/(b-a);
50     s2=(b-a)/(d-a);
51     return (c*s2+d*s1)/(s1+s2);
52 }
53 double sector_area(point a,point b,double r) {
54     double theta;
55     theta=atan2(a.y,a.x)-atan2(b.y,b.x);
56     while (theta <= 0) theta+=2*PI;
57     while (theta > 2*PI) theta-=2*PI;
58     theta=min(theta,2*PI-theta);
59     return r*r*theta/2;
60 }
61 void circle_cross_line(point o,double r,point a,point b,point ret[2],int &num) {
62     double ax=a.x-o.x;
63     double ay=a.y-o.y;
64     double bx=b.x-o.x;
65     double by=b.y-o.y;
66     double dx=bx-ax;
67     double dy=by-ay;
68     double A=dx*dx+dy*dy;
69     double B=2*(ax*dx+ay*dy);
70     double C=ax*ax+ay*ay-r*r;
71     double delta=B*B-4*A*C;
72     num=0;
73     if (dcmp(delta)>=0) {
74         double t1=(-B-sqrt(delta))/(2*A);
75         double t2=(-B+sqrt(delta))/(2*A);
76         if (dcmp(t1-1)<=0&&dcmp(t1)>=0)
77             ret[num++]=point(a.x+t1*dx,a.y+t1*dy);
78         if (dcmp(t2-1)<=0&&dcmp(t2)>=0)
79             ret[num++]=point(a.x+t2*dx,a.y+t2*dy);
80     }
81 }
82 double calc(point &a,point &b) {
83     bool ina,inb;
84     point ret[2];
85     int num;

```

```

86 if (O==a||O==b) return 0;
87 if (a==b) return 0;
88 ina=dcmp(a).abs()-R<=0;
89 inb=dcmp(b).abs()-R<=0;
90 if (ina) {
91     if (inb) {
92         return abs(a/b)/2.0;
93     }
94     else {
95         circle_cross_line(0,R,a,b,ret,num);
96         return abs(a/ret[num-1])/2.0+sector_area(ret[num-1],b,R);
97     }
98 }
99 else{
100     if (inb) {
101         circle_cross_line(0,R,b,a,ret,num);
102         return abs(b/ret[num-1])/2.0+sector_area(ret[num-1],a,R);
103     }
104     else {
105         circle_cross_line(0,R,a,b,ret,num);
106         if (num==2) {
107             return sector_area(a,ret[0],R)+abs(ret[0]/ret[1])/2.0+sector_area(
108                 ret[1],b,R);
109         }
110         else {
111             return sector_area(a,b,R);
112         }
113     }
114 }
115 int main() {
116     int i,x,y,T=0;
117     point u,v;
118     point p1,p2;
119     point mid;
120     while (scanf("%d%lf",&n,&k)!=EOF) {
121         for (i=1; i<=n; i++) {
122             scanf("%d%d",&x,&y);
123             p[i]=point(x,y);
124         }
125         scanf("%d%d",&x,&y);
126         u=point(x,y);
127         scanf("%d%d",&x,&y);
128         v=point(x,y);
129         p1=v+(u-v)*k/(1+k);
130         p2=v+(u-v)*k/(k-1);
131         mid=(p1+p2)/2;
132         R=(p1-p2).abs()/2;

```

```

133         for (i=1; i<=n; i++) {
134             p[i]=p[i]-mid;
135         }
136         p[n+1]=p[1];
137         ans=0;
138         for (i=1; i<=n; i++) {
139             int sig;
140             sig=dcmp(p[i]/p[i+1]);
141             double temp;
142             temp=calc(p[i],p[i+1]);
143             if (sig==0) ans+=temp;
144             else ans-=temp;
145         }
146         ans=abs(ans);
147         printf("Case %d: %.12f\n",++T,ans);
148     }
149     return 0;
150 }

```

9 hdu5384(ac)

```

1 #include<iostream>
2 #include<cstdio>
3 #include<cstring>
4 #include<vector>
5 #include<queue>
6 using namespace std;
7 int n,m;
8 char ch[20001];
9 string a[100001];
10 struct node{
11     int ch[26];
12     int fa;
13     int v;
14     int fail;
15     void init() {
16         fa=0;
17         v=0;
18         fail=0;
19         memset(ch,0,sizeof ch);
20     }
21 } T[200001];
22 int tot;
23 void add(int id) {
24     int i,j,l,u=0,y;
25     l=strlen(ch);
26     for (i=0; i<l; i++) {

```

```

27     y=ch[i]-'a';
28     if (!T[u].ch[y]) {
29         T[u].ch[y]=++tot;
30         T[tot].init();
31         T[tot].fa=u;
32     }
33     u=T[u].ch[y];
34 }
35 T[u].v++;
36 }
37 queue<int> q;
38 void ac() {
39     int i,j,k,v,fail;
40     while (!q.empty()) q.pop();
41     T[0].fail=-1;
42     q.push(0);
43     while (!q.empty()) {
44         i=q.front();
45         q.pop();
46         fail=T[i].fail;
47         if (fail!=-1) T[i].v+=T[fail].v;
48         for (j=0; j<26; j++)
49             if (T[i].ch[j]) {
50                 v=T[i].ch[j];
51                 if (fail!=-1) T[v].fail=0;
52                 else T[v].fail=T[fail].ch[j];
53                 q.push(v);
54             }
55         else if (fail!=-1) {
56             T[i].ch[j]=T[fail].ch[j];
57         }
58     }
59 }
60 void cal(int id) {
61     int i,j,k,l,u=0,v,ret=0;
62     l=a[id].length();
63     for (i=0; i<l; i++) {
64         j=a[id][i]-'a';
65         u=T[u].ch[j];
66         v=T[u].v;
67         ret+=v;
68     }
69     printf("%d\n",ret);
70 }
71 void print() {
72     int i,j,l,v;
73     for (i=0; i<=tot; i++) {
74         cout<<i<<": "<<endl;

```

```

75         for (j=0; j<26; j++)
76             if (T[i].ch[j]) cout<<char(j+'a')<<": "<<T[i].ch[j]<< " ";
77         cout<<endl;
78     }
79 }
80 int main() {
81     int __,i;
82     scanf("%d",&__);
83     while (__) {
84         scanf("%d%d",&n,&m);
85         for (i=1; i<=n; i++) {
86             scanf("%s",ch);
87             a[i]=ch;
88         }
89         tot=0;
90         T[0].init();
91         for (i=1; i<=m; i++) {
92             scanf("%s",ch);
93             add(i);
94         }
95         ac();
96         for (i=1; i<=n; i++) {
97             cal(i);
98         }
99     }
100     return 0;
101 }

```

10 poj2417(bsgs)

```

1 #include<iostream>
2 #include<cstdio>
3 #include<cstring>
4 #include<cmath>
5 #include<set>
6 #include<vector>
7 using namespace std;
8 typedef long long ll;
9 const int mod=100007;
10 int P,B,N;
11 int m;
12 int pow(int a,int b) {
13     int ret=1;
14     while (b) {
15         if (b&1) ret=(ll)ret*a%P;
16         a=(ll)a*a%P;
17         b>>=1;

```

```

18 }
19 return ret;
20 }
21 set<int> s;
22 vector<pair<int,int> > hs[100007];
23 set<int>::iterator it;
24 void add(int u,int id) {
25     int i,j,l;
26     j=u%mod;
27     l=hs[j].size();
28     for (i=0; i<l; i++)
29         if (hs[j][i].first==u) {
30             hs[j][i].second=id;
31             return;
32         }
33     hs[j].push_back(make_pair(u,id));
34     s.insert(j);
35 }
36 int ask(int u) {
37     int i,j,l;
38     j=u%mod;
39     l=hs[j].size();
40     for (i=0; i<l; i++)
41         if (hs[j][i].first==u) {
42             return hs[j][i].second;
43         }
44     return -1;
45 }
46 void BSGS() {
47     int i,j,k,t;
48     m=ceil(sqrt(P));
49     for (i=0,j=1; i<m; i++,j=(ll)j*B%P) {
50         if (j==N) {
51             printf("%d\n",i);
52             return;
53         }
54         add((ll)j*N%P,i);
55     }
56     for (i=1,k=j; i<=m; i++,k=(ll)k*j%P) {
57         t=ask(k);
58         if (t!=-1) {
59             t=i*m-t;
60             printf("%d\n",t);
61             return;
62         }
63     }
64     printf("no solution\n");
65 }

```

```

66 void init() {
67     for (it=s.begin(); it!=s.end(); it++) {
68         hs[*it].clear();
69     }
70     s.clear();
71 }
72 int main() {
73     s.clear();
74     while (scanf("%d%d%d",&P,&B,&N)!=EOF) {
75         init();
76         BSGS();
77     }
78     return 0;
79 }

```

11 tarjan

```

1 struct node{
2     int to,next;
3     bool flag;
4 } w[300001];
5 int h[100001],cnt=1;
6 void add(int x,int y) {
7     w[++cnt]=(node){y,h[x],0};
8     h[x]=cnt;
9     w[++cnt]=(node){x,h[y],0};
10    h[y]=cnt;
11 }
12 int dfn[100001];
13 int low[100001];
14 int tot;
15 void tarjan(int u,int fa) {
16     int j,v;
17     dfn[u]=low[u]=++tot;
18     for (v=h[u]; v; v=w[v].next) {
19         j=w[v].to;
20         if (j==fa) continue;
21         if (!dfn[j]) {
22             tarjan(j,u);
23             if (low[j]>dfn[u]) w[v].flag=1,w[v^1].flag=1;
24             low[u]=min(low[u],low[j]);
25         }
26         else low[u]=min(low[u],dfn[j]);
27     }
28 }

```