ACM/ICPC 代码模板库

南京大学 ACM/ICPC 集训队

目录

1 比赛配置

1.1 代码库校验和

```
01b4 # 代码库校验用于检查代码库录入是否正确，忽略
       每行的空白字符和注释(//)
44f9 # 使用方法: python checksum.py < 1001.cpp
4de6 # 输出: 每一行代码及其校验和(md5)
c502 import re, sys, hashlib
427e
b41f def digest_line(s):
d74e   return hashlib.md5(re.sub(r'\s|//.*',
         '', s)).hexdigest()[-4:]
427e
f7db for line in sys.stdin.read().strip().
       split("\n"):
f335   print digest_line(line), line
```

1.2 vimrc

```
3ff1 color evening
7db5 set number
7232 set cindent
427e
427e
427e
6cb2 function HomeBind(offset)
7d5c   let cursor=getpos('.')
bda7   let s0=getline(line('.'))
1903   let s1=substitute(s0, "^\\s\\+", "",
         "")
7f8d   let x=len(s0)-len(s1)+1
2b1d   if col('.') == x-a:offset
0437     let x=1
400b   endif
d7af   call setpos('.', [cursor[0], cursor[1],
          x, cursor[3]])
f298 endfunction
1a4a imap <silent> <Home> <Esc>:call HomeBind
       (1)<cr>i
73c9 nmap <silent> <Home> :call HomeBind(0)<cr
       >
b506 vmap <silent> <Home> <Esc>:call HomeBind
       (1)<cr>
```

1.3 代码库校验和

```
01b4 # 代码库校验用于检查代码库录入是否正确，忽略
       每行的空白字符和注释(//)
44f9 # 使用方法: python checksum.py < 1001.cpp
4de6 # 输出: 每一行代码及其校验和(md5)
c502 import re, sys, hashlib
427e
def digest_line(s):                                  b41f
  return hashlib.md5(re.sub(r'\s|//.*', '            d74e
    ', s)).hexdigest()[-4:]
                                                     427e
for line in sys.stdin.read().strip().                f7db
  split("\n"):
  print digest_line(line), line                      f335
```

1.4 vim 配置文件

```
# vimrc 配置文件                                     44ed
set nocompatible                                     914c
set number                                           7db5
set ruler                                            57b2
set showcmd                                          9832
set autoindent                                       e416
set cindent                                          7232
set smartindent                                      740c
set shiftwidth=4                                     5913
```

1.5 外挂

```
// 调栈空间                                           427e
const int N_MAX = 10000000;                          08e0
static int stack[N_MAX * 5], bak;                    772b
asm __volatile__                                     68df
  (                                                  bbf4
    "movl %%esp, %0;"                                4d52
    "movl %1, %%esp;":                               22d3
    "=g"(bak):                                       7f26
    "g"(stack + N_MAX * 5 - 1):                      5bf1
  );                                                 fe62
                                                     427e
// IO 外挂                                            427e
#define BUFSIZE 20000000                             6540
char buf[BUFSIZE], *pt = buf;                        eb51
#define scan(t) \                                    45da
{ \                                                  c2a7
  t = 0; \                                           f000
  while (!((*pt) >= '0' && (*pt) <= '9'))            2760
     pt ++; \
  while (((*pt) >= '0' && (*pt) <= '9'))             7181
    t = t * 10 + (*(pt ++)) - '0'; \
}                                                    95cf
                                                     427e
 int main()                                          299c
{                                                    4506
 fread(buf, 1, BUFSIZE, stdin);                      486b
scan(N);                                             0756
}                                                    95cf
                                                     427e
// C++ 编译器(VS) 调栈空间                            427e
#pragma comment(linker,"/STACK                       85cb
  :102400000,102400000")
```

2 二维计算几何

2.1 定义

```
652e #define eps 1e-8
c1b0 #define fabs(x) ((x) > 0? (x): -(x))
0102 #define zero(x) (fabs(x) < eps)
12d8 #define _sign(x) ((x)>eps?1:((x)<-eps
       ?2:0))
dca2 #define sqr(x) ((x)*(x))
418f #define MAXN 1000
8d0a #define offset 10000
13f1 const double pi=acos(-1);
427e
427e
427e // 点的定义
9704 struct point{
082e   int index;
98c9   double ang;
d0aa   double x, y;
52a2   point(){x = 0;y = 0;}
df98   point(double sx, double sy){
e87b     x = sx;
d22b     y = sy;
95cf   }
4f13   void read(){
dab2     scanf("%lf %lf ", &x, &y);
95cf   }
a7a6   bool operator <(const point &b)const{
73b2     if (b.x == x) return y < b.y;
66d1     return x < b.x;
95cf   }
7b0b   point operator - (const point &b)const
         {
f32c     point a;
d53d     a.x = x - b.x;
5365     a.y = y - b.y;
5ffd     return a;
95cf   }
e254   point operator + (const point &b)const{
f32c     point a;
7683     a.x = x + b.x;
70a0     a.y = y + b.y;
5ffd     return a;
95cf   }
14f6   point operator / (const double &c)const
         {
f32c     point a;
225c     a.x = x / c;
414d     a.y = y / c;
5ffd     return a;
95cf   }
d466   point operator * (const double &c)const
         {
f32c     point a;
7aa6     a.x = x * c;
9a5c     a.y = y * c;
5ffd     return a;
95cf   }
12ba   bool operator == (const point &p) const
          {
e89b     return zero(x - p.x)&&zero(y - p.y);
95cf   }
daed   friend ostream& operator << (ostream &
         out, const point &a);
329b };
f71b typedef const point CP;
05c6 ostream& operator << (ostream &out, const
        point &a){
df9c   out<<a.x<<' '<<a.y;
d324   return out;
95cf }
eb7f bool cmp(const point &p1, const point &p2
       ){
84df   return p1.ang < p2.ang;
95cf }
427e
427e // 线定义, 使用< 进行极角排序之前需要对所有
       线段调用getang 函数
bda3 struct line{
98c9   double ang;
a334   point a, b;line(){};
5ae1   line(const point &p1, const point &p2){
0fa8     a = p1;
ce41     b = p2;
95cf   }
cfb1   bool operator < (const line &y)const{
8e85     if (zero(ang - y.ang))
37ec       return (xmult(a, y.b, y.a) < 0);
3b83     return ang < y.ang;
95cf   }
1d99   void getang(){
f061     ang = atan2(b.y - a.y, b.x - a.x);
95cf   }
4c7b   friend ostream& operator << (ostream &
         out, const line &a);
329b };
cdd9 ostream& operator << (ostream &out, const
        line &a){
1d6a   out<<a.a<<' '<<a.b<<' '<<a.ang;
d324   return out;
95cf }
a5b5 typedef const line CL;
427e // 圆定义
2f47 struct circle{
0c09   double r;
67a8    point c;
c0ad    circle(){};
614e    circle(const point &p, double x){
c23d      c = p;
d0c6      x = r;
95cf    }
a9b1    friend ostream& operator << (ostream &
          out, const circle &a);
329b  };
b802  ostream& operator << (ostream &out, const
         circle &a){
d469    out<<a.c<<' '<<a.r;
d324    return out;
95cf  }
e519  typedef const circle CC;
```

2.2 定义

```
652e  #define eps 1e–8
c1b0  #define fabs(x) ((x) > 0? (x): –(x))
0102  #define zero(x) (fabs(x) < eps)
12d8  #define _sign(x) ((x)>eps?1:((x)<–eps
        ?2:0))
418f  #define MAXN 1000
8d0a  #define offset 10000
dca2  #define sqr(x) ((x)*(x))
13f1  const double pi=acos(–1);
427e
427e
427e  // 点的定义
9704  struct point{
082e    int index;
98c9    double ang;
d0aa    double x, y;
52a2    point(){x = 0;y = 0;}
df98    point(double sx, double sy){
e87b      x = sx;
d22b      y = sy;
95cf    }
a7a6    bool operator <(const point &b)const{
73b2      if (b.x == x) return y < b.y;
66d1      return x < b.x;
95cf    }
7b0b    point operator – (const point &b)const
          {
f32c      point a;
d53d      a.x = x – b.x;
5365      a.y = y – b.y;
5ffd      return a;
95cf    }
e254    point operator + (const point &b)const{
f32c      point a;
7683      a.x = x + b.x;
70a0      a.y = y + b.y;
```

```
    return a;                                    5ffd
  }                                              95cf
  point operator / (const double &c)const        14f6
    {
    point a;                                     f32c
    a.x = x / c;                                 225c
    a.y = y / c;                                 414d
    return a;                                    5ffd
  }                                              95cf
  point operator * (const double &c)const        d466
    {
    point a;                                     f32c
    a.x = x * c;                                 7aa6
    a.y = y * c;                                 9a5c
    return a;                                    5ffd
  }                                              95cf
  bool operator == (const point &p) const        12ba
     {
    return zero(x – p.x)&&zero(y – p.y);         e89b
  }                                              95cf
  friend ostream& operator << (ostream &         daed
    out, const point &a);
};                                               329b
ostream& operator << (ostream &out, const        05c6
   point &a){
  out<<a.x<<' '<<a.y;                            df9c
  return out;                                    d324
}                                                95cf
bool cmp(const point &p1, const point &p2        eb7f
  ){
  return p1.ang < p2.ang;                        84df
}                                                95cf
                                                 427e
// 线定义, 使用< 进行极角排序之前需要对所有         427e
  线段调用getang 函数
struct line{                                     bda3
  double ang;                                    98c9
  point a, b;line(){};                           a334
  line(const point &p1, const point &p2){        5ae1
    a = p1;                                      0fa8
    b = p2;                                      ce41
  }                                              95cf
  bool operator < (const line &y)const{          cfb1
    if (zero(ang – y.ang))                       8e85
      return (xmult(a, y.b, y.a) < 0);           37ec
    return ang < y.ang;                          3b83
  }                                              95cf
  void getang(){                                 1d99
    ang = atan2(b.y – a.y, b.x – a.x);           f061
  }                                              95cf
  friend ostream& operator << (ostream &         4c7b
    out, const line &a);
};                                               329b
```

```
cdd9 ostream& operator << (ostream &out, const
        line &a){
1d6a   out<<a.a<<' '<<a.b<<' '<<a.ang;
d324   return out;
95cf }
427e
427e // 圆定义
2f47 struct circle{
0c09   double r;
67a8   point c;
c0ad   circle(){};
614e   circle(const point &p, double x){
c23d     c = p;
d0c6     x = r;
95cf   }
a9b1   friend ostream& operator << (ostream &
         out, const circle &a);
329b };
b802 ostream& operator << (ostream &out, const
        circle &a){
d469   out<<a.c<<' '<<a.r;
d324   return out;
95cf }
```

2.3 点与线

```
427e // 计算cross product (P1-P0)x(P2-P0)
9060 double xmult(CP &p1,CP &p2,CP &p0){
a01c   return (p1.x-p0.x)*(p2.y-p0.y)-(p2.x-p0
         .x)*(p1.y-p0.y);
95cf }
26f9 double xmult(double x1,double y1,double
       x2,double y2,double x0,double y0){
c71e   return (x1-x0)*(y2-y0)-(x2-x0)*(y1-y0);
95cf }
0209 double xmult(CP &v1, CP &v2){
12a3   return v1.x * v2.y - v2.x * v1.y;
95cf }
427e
427e // 计算dot product (P1-P0).(P2-P0)
dbb2 double dmult(CP &p1,CP &p2,CP &p0){
6b50   return (p1.x-p0.x)*(p2.x-p0.x)+(p1.y-p0
         .y)*(p2.y-p0.y);
95cf }
8c06 double dmult(double x1,double y1,double
       x2,double y2,double x0,double y0){
4103   return (x1-x0)*(x2-x0)+(y1-y0)*(y2-y0);
95cf }
cb7a double dmult(CP &v1, CP &v2){
d871   return v1.x * v2.x + v1.y * v2.y;
95cf }
427e
427e // 计算向量v 的长度
bd72 double len(CP &v){
```

```
  return sqrt((v.x * v.x) + (v.y * v.y));   7c36
}                                           95cf
                                            427e
// 两点距离                                  427e
double dis(CP &p1,CP &p2){                  21ca
  return sqrt((p1.x-p2.x)*(p1.x-p2.x)+(p1   b08a
    .y-p2.y)*(p1.y-p2.y));
}                                           95cf
double dis(double x1,double y1,double x2,   8046
  double y2){
  return sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1   9f44
    -y2));
}                                           95cf
double dis2(CP &p1,CP &p2){                 1627
  return (p1.x-p2.x)*(p1.x-p2.x)+(p1.y-p2   3eaf
    .y)*(p1.y-p2.y);
}                                           95cf
double dis2(double x1,double y1,double x2   a880
  ,double y2){
  return (x1-x2)*(x1-x2)+(y1-y2)*(y1-y2);   5819
}                                           95cf
                                            427e
// 判三点共线                                427e
int dots_inline(CP &p1,CP &p2,CP &p3){      68d7
  return zero(xmult(p1,p2,p3));             20b6
}                                           95cf
int dots_inline(double x1,double y1,        a6a7
  double x2,double y2,double x3,double y3
  ){
  return zero(xmult(x1,y1,x2,y2,x3,y3));    fc4c
}                                           95cf
                                            427e
// 判点是否在线段上, 包括端点                 427e
int dot_online_in(CP &p,CL &l){             23f7
  return zero(xmult(p,l.a,l.b))&&(l.a.x-p   d7f1
    .x)*(l.b.x-p.x)<eps&&(l.a.y-p.y)*(l.b
    .y-p.y)<eps;
}                                           95cf
int dot_online_in(CP &p,CP &l1,CP &l2){     a66d
  return zero(xmult(p,l1,l2))&&(l1.x-p.x)   9f13
    *(l2.x-p.x)<eps&&(l1.y-p.y)*(l2.y-p.y
    )<eps;
}                                           95cf
int dot_online_in(double x,double y,        86fb
  double x1,double y1,double x2,double y2
  ){
  return zero(xmult(x,y,x1,y1,x2,y2))&&(    5e8a
    x1-x)*(x2-x)<eps&&(y1-y)*(y2-y)<eps;
}                                           95cf
                                            427e
// 判点是否在线段上不包括端点,                427e
int dot_online_ex(CP &p,CL &l){             84f5
  return dot_online_in(p,l)&&(!zero(p.x-l   289e
         .a.x)||!zero(p.y-l.a.y))&&(!zero(p.x-
         l.b.x)||!zero(p.y-l.b.y));
95cf }
a73f int dot_online_ex(CP &p,CP &l1,CP &l2){
82af   return dot_online_in(p,l1,l2)&&(!zero(p
         .x-l1.x)||!zero(p.y-l1.y))&&(!zero(p.
         x-l2.x)||!zero(p.y-l2.y));
95cf }
0181 int dot_online_ex(double x,double y,
       double x1,double y1,double x2,double y2
       ){
58c5   return dot_online_in(x,y,x1,y1,x2,y2)
         &&(!zero(x-x1)||!zero(y-y1))&&(!zero(
         x-x2)||!zero(y-y2));
95cf }
427e
427e // 判两点在线段同侧, 点在线段上返回0
ea31 int same_side(CP &p1,CP &p2,CL &l){
d345   return xmult(l.a,p1,l.b)*xmult(l.a,p2,l
         .b)>eps;
95cf }
8a3c int same_side(CP &p1,CP &p2,CP &l1,CP &l2
       ){
14cc   return xmult(l1,p1,l2)*xmult(l1,p2,l2)>
         eps;
95cf }
427e
427e // 判两点在线段异侧, 点在线段上返回0
b5f2 int opposite_side(CP &p1,CP &p2,CL &l){
95bc   return xmult(l.a,p1,l.b)*xmult(l.a,p2,l
         .b)<-eps;
95cf }
d050 int opposite_side(CP &p1,CP &p2,CP &l1,CP
      &l2){
de06   return xmult(l1,p1,l2)*xmult(l1,p2,l2)
        <-eps;
95cf }
427e
427e // 判两直线平行
efdb int parallel(CL &u,CL &v){
92d6   return zero((u.a.x-u.b.x)*(v.a.y-v.b.y)
         -(v.a.x-v.b.x)*(u.a.y-u.b.y));
95cf }
3419 int parallel(CP &u1,CP &u2,CP &v1,CP &v2)
       {
4806   return zero((u1.x-u2.x)*(v1.y-v2.y)-(v1
         .x-v2.x)*(u1.y-u2.y));
95cf }
427e
427e // 判两直线垂直
7e98 int perpendicular(CL &u,CL &v){
980d   return zero((u.a.x-u.b.x)*(v.a.x-v.b.x)
         +(u.a.y-u.b.y)*(v.a.y-v.b.y));
```

```
}                                            95cf
int perpendicular(CP &u1,CP &u2,CP &v1,CP    518c
   &v2){
  return zero((u1.x-u2.x)*(v1.x-v2.x)+(u1    6143
    .y-u2.y)*(v1.y-v2.y));
}                                            95cf
                                             427e
// 判两线段相交, 包括端点和部分重合           427e
int intersect_in(CL &u,CL &v){               779d
  if (!dots_inline(u.a,u.b,v.a)||!           e887
    dots_inline(u.a,u.b,v.b))
    return !same_side(u.a,u.b,v)&&!          d666
      same_side(v.a,v.b,u);
  return dot_online_in(u.a,v)||              efeb
    dot_online_in(u.b,v)||dot_online_in(v
    .a,u)||dot_online_in(v.b,u);
}                                            95cf
int intersect_in(CP &u1,CP &u2,CP &v1,CP     2882
  &v2){
  if (!dots_inline(u1,u2,v1)||!              8fcb
    dots_inline(u1,u2,v2))
    return !same_side(u1,u2,v1,v2)&&!        4b79
      same_side(v1,v2,u1,u2);
  return dot_online_in(u1,v1,v2)||           cc40
    dot_online_in(u2,v1,v2)||
    dot_online_in(v1,u1,u2)||
    dot_online_in(v2,u1,u2);
}                                            95cf
                                             427e
// 判两线段相交, 不包括端点和部分重合         427e
int intersect_ex(CL &u,CL &v){               cc64
  return opposite_side(u.a,u.b,v)&&          bedb
    opposite_side(v.a,v.b,u);
}                                            95cf
int intersect_ex(CP &u1,CP &u2,CP &v1,CP     5c8b
  &v2){
  return opposite_side(u1,u2,v1,v2)&&        70ea
    opposite_side(v1,v2,u1,u2);
}                                            95cf
                                             427e
// 计算两直线交点, 注意事先判断直线是否平行!  427e
// 线段交点请另外判线段相交, 同时还是要判断   427e
  是否平行
point intersection(CL &u,CL &v){             afec
  point ret=u.a;                             39f9
  double t=((u.a.x-v.a.x)*(v.a.y-v.b.y)-(    273a
    u.a.y-v.a.y)*(v.a.x-v.b.x))
      /((u.a.x-u.b.x)*(v.a.y-v.b.y)-(u.a.    9cb3
        y-u.b.y)*(v.a.x-v.b.x));
  ret.x+=(u.b.x-u.a.x)*t;                    1143
  ret.y+=(u.b.y-u.a.y)*t;                    12e9
  return ret;                                ee0f
}                                            95cf
```

```
fbb7 point intersection(CP &u1,CP &u2,CP &v1,
       CP &v2){
f0a0   point ret=u1;
a7db   double t=((u1.x-v1.x)*(v1.y-v2.y)-(u1.y
         -v1.y)*(v1.x-v2.x))
16f0       /((u1.x-u2.x)*(v1.y-v2.y)-(u1.y-u2.
           y)*(v1.x-v2.x));
a1f8   ret.x+=(u2.x-u1.x)*t;
fa1b   ret.y+=(u2.y-u1.y)*t;
ee0f   return ret;
95cf }
427e // 求给定线段的中垂线
1cc5 line pbline(CL &l){
671e   line ret;ret.a = (l.a + l.b) / 2;
d118   double a = l.b.x - l.a.x, b = l.b.y - l
         .a.y;
a2fc   double c = (l.a.y - l.b.y) * ret.a.y +
         (l.a.x - l.b.x) * ret.a.x;
c99e   if (!zero(a)){
03b7     ret.b.y = 0;ret.b.x = -c / a;
effa     if (zero(dis(ret.a, ret.b))){
dca4       ret.b.y = 1e10;ret.b.x = - (c - b *
              ret.b.y) / a;
95cf     }
8e2e   }else{
08d6     ret.b.x = 0.0;ret.b.y = -c / b;
effa     if (zero(dis(ret.a, ret.b))){
3b6b       ret.b.x = 1e10;ret.b.y = -(c - a *
             ret.b.x) / b;
95cf     }
95cf   }
ee0f   return ret;
95cf }
427e
427e // 点到直线上的最近点
27b7 point ptoline(CP &p,CL &l){
1960   point t=p;
c7d0   t.x+=l.a.y-l.b.y,t.y+=l.b.x-l.a.x;
2f7b   return intersection(p,t,l.a,l.b);
95cf }
b7a1 point ptoline(CP &p,CP &l1,CP &l2){
1960   point t=p;
e925   t.x+=l1.y-l2.y,t.y+=l2.x-l1.x;
60f2   return intersection(p,t,l1,l2);
95cf }
427e
427e // 点到直线距离
a82a double disptoline(CP &p,CL &l){
9546   return fabs(xmult(p,l.a,l.b))/dis(l.a,l
         .b);
95cf }
e8c8 double disptoline(CP &p,CP &l1,CP &l2){
ccfa   return fabs(xmult(p,l1,l2))/dis(l1,l2);
95cf }
83c9 double disptoline(double x,double y,
       double x1,double y1,double x2,double y2
       ){
d1a4   return fabs(xmult(x,y,x1,y1,x2,y2))/dis
         (x1,y1,x2,y2);
95cf }
427e
427e // 点到线段上的最近点
59a9 point ptoseg(CP &p,CL &l){
1960   point t=p;
c7d0   t.x+=l.a.y-l.b.y,t.y+=l.b.x-l.a.x;
c788   if (xmult(l.a,t,p)*xmult(l.b,t,p)>eps)
4763     return dis(p,l.a)<dis(p,l.b)?l.a:l.b;
2f7b   return intersection(p,t,l.a,l.b);
95cf }
0e5b point ptoseg(CP &p,CP &l1,CP &l2){
1960   point t=p;
e925   t.x+=l1.y-l2.y,t.y+=l2.x-l1.x;
b478   if (xmult(l1,t,p)*xmult(l2,t,p)>eps)
9083     return dis(p,l1)<dis(p,l2)?l1:l2;
60f2   return intersection(p,t,l1,l2);
95cf }
427e
427e // 点到线段距离
d9a8 double disptoseg(CP &p,CL &l){
1960   point t=p;
c7d0   t.x+=l.a.y-l.b.y,t.y+=l.b.x-l.a.x;
c788   if (xmult(l.a,t,p)*xmult(l.b,t,p)>eps)
9b41     return dis(p,l.a)<dis(p,l.b)?dis(p,l.
           a):dis(p,l.b);
9546   return fabs(xmult(p,l.a,l.b))/dis(l.a,l
         .b);
95cf }
0b5b double disptoseg(CP &p,CP &l1,CP &l2){
1960   point t=p;
e925   t.x+=l1.y-l2.y,t.y+=l2.x-l1.x;
b478   if (xmult(l1,t,p)*xmult(l2,t,p)>eps)
234b     return dis(p,l1)<dis(p,l2)?dis(p,l1):
           dis(p,l2);
ccfa   return fabs(xmult(p,l1,l2))/dis(l1,l2);
95cf }
427e
427e // 线段到线段距离，事先判断相交情况
d24d double dissegtoseg(CL &l1, CL &l2){
37e7   return min(min(disptoseg(l1.a, l2),
         disptoseg(l1.b, l2)), min(disptoseg(
         l2.a, l1), disptoseg(l2.b, l1)));
95cf }
d0fd double dissegtoseg(CP &l1a, CP &l1b, CP &
       l2a, CP &l2b){
c92c   return min(min(disptoseg(l1a, l2a, l2b)
         , disptoseg(l1b, l2a, l2b)), min(
        disptoseg(l2a, l1a, l1b), disptoseg(
        l2b, l1a, l1b)));
95cf  }
427e
427e  // 矢量V 以P 为顶点逆时针旋转angle 并放
        大scale 倍
546e  point rotate(point v,point p,double angle
        ,double scale){
15e2    point ret=p;
cee4    v.x-=p.x,v.y-=p.y;
2350    p.x=scale*cos(angle);
6a5a    p.y=scale*sin(angle);
2bf2    ret.x+=v.x*p.x-v.y*p.y;
0ad4    ret.y+=v.x*p.y+v.y*p.x;
ee0f    return ret;
95cf  }
427e
427e  // p 在新坐标系O(I,e1,e2) 中的坐标
f6d4  point rotate(CP &p, CP &I, CP &e1, CP &e2
        ){
22ef    point p2;
495d    p2.x = I.x + e1.x * p.x + e1.y * p.y;
84f1    p2.y = I.y + e2.x * p.x + e2.y * p.y;
7a0c    return p2;
427e
95cf  }
427e
427e  //p 点绕原点按逆时针旋转angle
3f0e  point rotate(CP &p, double angle){
3e07    point e1, e2, I;
1f1c    e1.x = cos(angle);e1.y = -sin(angle);
ed21    e2.x = -e1.y;e2.y = e1.x;
085d    I.x = 0;I.y = 0;
e586    return rotate(p, I, e1, e2);
95cf  }
427e
427e  // 返回值[0, 4), 正比向量v1 到向量v2 的顺时
        针旋转角度
452a  double angle(CP &v1, CP &v2){
94c9    double cosa = dmult(v1, v2) / len(v1) /
           len(v2);cosa = 1 - cosa;
53df    if (xmult(v1, v2) < 0) cosa = 4 - cosa;
          return cosa;
95cf  }
8cb3  double angle(CP &v1, CP &a, CP &b){
ce1f    return angle(v1, b - a);
95cf  }
427e  // 向量a-c 到b-c 的顺时针度数
aab2  double angle(CP &a, CP &b, CP &c){
427e    //return angle(a - b, b - c);
76a4    double cosa = dmult(a, b, c) / dis(a, c
          ) / dis(b, c);cosa = 1 - cosa;
c807    if (xmult(a, b, c) < 0) cosa = 4 - cosa
```

```
    ;return cosa;
}                                              95cf
```

2.4 三角形

```
// 外心                                        427e
point circumcenter(CP &a,CP &b,CP &c){         878f
  line u,v;                                    7173
  u.a.x=(a.x+b.x)/2;                           2ecf
  u.a.y=(a.y+b.y)/2;                           5a23
  u.b.x=u.a.x-a.y+b.y;                         ee31
  u.b.y=u.a.y+a.x-b.x;                         77b1
  v.a.x=(a.x+c.x)/2;                           5396
  v.a.y=(a.y+c.y)/2;                           8492
  v.b.x=v.a.x-a.y+c.y;                         abe9
  v.b.y=v.a.y+a.x-c.x;                         63ff
  return intersection(u,v);                    ed5d
}                                              95cf
                                               427e
// 内心                                        427e
point incenter(CP &a,CP &b,CP &c){             6bd8
  line u,v;                                    7173
  double m,n;                                  1a7c
  u.a=a;                                       2aec
  m=atan2(b.y-a.y,b.x-a.x);                    cd0a
  n=atan2(c.y-a.y,c.x-a.x);                    4211
  u.b.x=u.a.x+cos((m+n)/2);                    af8c
  u.b.y=u.a.y+sin((m+n)/2);                    b80b
  v.a=b;                                       93b1
  m=atan2(a.y-b.y,a.x-b.x);                    da57
  n=atan2(c.y-b.y,c.x-b.x);                    0a23
  v.b.x=v.a.x+cos((m+n)/2);                    3d04
  v.b.y=v.a.y+sin((m+n)/2);                    b99c
  return intersection(u,v);                    ed5d
}                                              95cf
                                               427e
// 垂心                                        427e
point perpencenter(CP &a,CP &b,CP &c){         8ae2
  line u,v;                                    7173
  u.a=c;                                       1ecd
  u.b.x=u.a.x-a.y+b.y;                         ee31
  u.b.y=u.a.y+a.x-b.x;                         77b1
  v.a=b;                                       93b1
  v.b.x=v.a.x-a.y+c.y;                         abe9
  v.b.y=v.a.y+a.x-c.x;                         63ff
  return intersection(u,v);                    ed5d
}                                              95cf
                                               427e
// 重心                                        427e
// 到三角形三顶点距离的平方和最小的点          427e
// 三角形内到三边距离之积最大的点              427e
point barycenter(CP &a,CP &b,CP &c){           81bd
  line u,v;                                    7173
  u.a.x=(a.x+b.x)/2;                           2ecf
5a23   u.a.y=(a.y+b.y)/2;
fad0   u.b=c;
5396   v.a.x=(a.x+c.x)/2;
8492   v.a.y=(a.y+c.y)/2;
f28e   v.b=b;
ed5d   return intersection(u,v);
95cf }
427e
427e // 费马点
427e // 到三角形三顶点距离之和最小的点
b0a9 point fermentpoint(CP &a,CP &b,CP &c){
03f5   point u,v;
c4d2   double step=fabs(a.x)+fabs(a.y)+fabs(b.
         x)+fabs(b.y)+fabs(c.x)+fabs(c.y);
c8ed   int i,j,k;
1eb8   u.x=(a.x+b.x+c.x)/3;
9137   u.y=(a.y+b.y+c.y)/3;
7bf1   while (step>1e-10)
0fcc     for (k=0;k<10;step/=2,k++)
13ea       for (i=-1;i<=1;i++)
9366         for (j=-1;j<=1;j++){
6701           v.x=u.x+step*i;
61f3           v.y=u.y+step*j;
7b0c           if (dis(u,a)+dis(u,b)+dis(u,c)>
                 dis(v,a)+dis(v,b)+dis(v,c))
c013             u=v;
95cf         }
81b0   return u;
95cf }
```

2.5 多边形

```
427e // 判定凸多边形，顶点按顺时针或逆时针给
       出，允许相邻边共线
7a16 int is_convex(int n,point* p){
53c4   int i,s[3]={1,1,1};
cb40   for (i=0;i<n&&s[1]|s[2];i++)
1a26     s[_sign(xmult(p[(i+1)%n],p[(i+2)%n],p
           [i]))]=0;
e8ad   return s[1]|s[2];
95cf }
427e
427e // 判定凸多边形，顶点按顺时针或逆时针给
       出，不允许相邻边共线
eba7 int is_convex_v2(int n,point* p){
53c4   int i,s[3]={1,1,1};
ae7f   for (i=0;i<n&&s[0]&&s[1]|s[2];i++)
1a26     s[_sign(xmult(p[(i+1)%n],p[(i+2)%n],p
           [i]))]=0;
e92f   return s[0]&&s[1]|s[2];
95cf }
427e
427e // 判点在凸多边形内或多边形边上，顶点按顺时
       针或逆时针给出
int inside_convex(CP &q,int n,point* p){     ea98
  int i,s[3]={1,1,1};                        53c4
  for (i=0;i<n&&s[1]|s[2];i++)               cb40
    s[_sign(xmult(p[(i+1)%n],q,p[i]))]=0;    8fa5
  return s[1]|s[2];                          e8ad
}                                            95cf
                                             427e
// 判点在凸多边形内，顶点按顺时针或逆时针给      427e
  出，在多边形边上返回0
int inside_convex_v2(CP &q,int n,point* p    9e0e
  ){
  int i,s[3]={1,1,1};                        53c4
  for (i=0;i<n&&s[0]&&s[1]|s[2];i++)         ae7f
    s[_sign(xmult(p[(i+1)%n],q,p[i]))]=0;    8fa5
  return s[0]&&s[1]|s[2];                    e92f
}                                            95cf
                                             427e
// 判点在任意多边形内顶点按顺时针或逆时针给      427e
  出，
// on_edge 表示点在多边形边上时的返回          427e
  值,offset 为多边形坐标上限
int inside_polygon(CP &q,int n,point* p,     78bd
  int on_edge=1){
  point q2;                                  af1a
  int i=0,count;                             1adf
  while (i<n)                                22c1
    for (count=i=0,q2.x=rand()+offset,q2.    ea47
      y=rand()+offset;i<n;i++)
      if (zero(xmult(q,p[i],p[(i+1)%n]))     960c
        &&(p[i].x-q.x)*(p[(i+1)%n].x-q.x)
        <eps&&(p[i].y-q.y)*(p[(i+1)%n].y-
        q.y)<eps)
        return on_edge;                      163a
      else if (zero(xmult(q,q2,p[i])))       19b2
        break;                               6173
      else if (xmult(q,p[i],q2)*xmult(q,p    f875
        [(i+1)%n],q2)<-eps&&xmult(p[i],q,
        p[(i+1)%n])*xmult(p[i],q2,p[(i+1)
        %n])<-eps)
        count++;                             45d2
  return count&1;                            9103
}                                            95cf
                                             427e
// 判线段在任意多边形内，顶点按顺时针或逆时      427e
  针给出，与边界相交返回1
int inside_polygon(CP &l1,CP &l2,int n,      2e50
  point* p){
  point t[MAXN],tt;                          b779
  int i,j,k=0;                               ff68
  if (!inside_polygon(l1,n,p)||!             535b
    inside_polygon(l2,n,p))
    return 0;                                7021
  for (i=0;i<n;i++)                          2dbf
```

```
6a8f     if (opposite_side(l1,l2,p[i],p[(i+1)%
           n])&&opposite_side(p[i],p[(i+1)%n],
           l1,l2))
7021       return 0;
d6e2     else if (dot_online_in(l1,p[i],p[(i
           +1)%n]))
29be       t[k++]=l1;
cbea     else if (dot_online_in(l2,p[i],p[(i
           +1)%n]))
3d75       t[k++]=l2;
1abb     else if (dot_online_in(p[i],l1,l2))
8b2e       t[k++]=p[i];
a83a   for (i=0;i<k;i++)
e784     for (j=i+1;j<k;j++){
2874       tt.x=(t[i].x+t[j].x)/2;
abed       tt.y=(t[i].y+t[j].y)/2;
4b25       if (!inside_polygon(tt,n,p))
7021         return 0;
95cf     }
7459   return 1;
95cf }
427e
427e // 多边形重心
3a12 point barycenter(int n,point* p){
54ec   point ret,t;
3444   double t1=0,t2;
a0f7   int i;
5b97   ret.x=ret.y=0;
440f   for (i=1;i<n-1;i++)
4241     if (fabs(t2=xmult(p[0],p[i],p[i+1]))>
           eps){
c812       t=barycenter(p[0],p[i],p[i+1]);
0f3e       ret.x+=t.x*t2;
531e       ret.y+=t.y*t2;
1ea2       t1+=t2;
95cf     }
63aa   if (fabs(t1)>eps)
a16e     ret.x/=t1,ret.y/=t1;
ee0f   return ret;
95cf }
427e
427e // 将多边形沿l1,l2 确定的直线切割在side 侧
       切割
427e // 保证l1,l2,side 不共线
3a0f void polygon_cut(int& n,point* p,CP &l1,
       CP &l2,CP &side){
7368   point pp[100];
9894   int m=0,i;
ee09   for (i=0;i<n;i++){
2f5d     if (same_side(p[i],side,l1,l2))
33c3       pp[m++]=p[i];
226e     if (!same_side(p[i],p[(i+1)%n],l1,l2)
           &&!(zero(xmult(p[i],l1,l2))&&zero(
```

```
      xmult(p[(i+1)%n],l1,l2))))
      pp[m++]=intersection(p[i],p[(i+1)%n       4119
        ],l1,l2);
  }                                             95cf
  for (n=i=0;i<m;i++)                           dea9
    if (!i||!zero(pp[i].x-pp[i-1].x)||!         08e5
      zero(pp[i].y-pp[i-1].y))
      p[n++]=pp[i];                             0d6a
  if (zero(p[n-1].x-p[0].x)&&zero(p[n-1].       f09f
    y-p[0].y))
    n--;                                        61b6
  if (n<3)                                      4046
    n=0;                                        91c9
}                                               95cf
                                                427e
// 求平行于v 的所有射线中，穿过的凸包中最左           427e
  边的点的坐标
// 凸包点按顺时针给出                               427e
point vector_throw_convex(int n, point *        e405
  convex, CP &v){
  int s = 0;double as = angle(v, convex[s       9c56
    ], convex[(s + 1) % n]);
  int t = n - 1;double at = angle(v,            c116
    convex[t], convex[(t + 1) % n]);
  while (s < t){                                bed9
    if (as >= at){s = t;break;}int mid =        a934
      (s + t + 1) / 2;
    double amid = angle(v, convex[mid],         ef17
      convex[(mid + 1) % n]);
    if (amid <= as){s = mid;as = amid;}         b31e
    else{t = mid - 1;at = angle(v, convex       8674
      [t], convex[(t + 1) % n]);}
  }                                             95cf
  return convex[(s + 1) % n];                   605e
}                                               95cf
                                                427e
                                                427e
// 求直线l1 是否穿过凸包，凸包按顺时针给出，         427e
  返回是否穿过
// p 储存凸包内l1 共线的某点                        427e
bool line_throw_convex(int n, point *           18a9
  convex, CL &l, point &p){
  point p1 = vector_throw_convex(n,             91c3
    convex, l.a - l.b);
  point p2 = vector_throw_convex(n,             a453
    convex, l.b - l.a);
  line l2(p1, p2);p = intersection(l, l2)       f6ad
    ;
  if (dot_online_in(p, l2)) return true;        56e0
    return false;
}                                               95cf
                                                427e
// 求射线是否穿过凸包，凸包按顺时针给出，返          427e
```

```
     回是否穿过
427e // p 储存凸包内l1 共线的某点
162c bool ray_throw_convex(int n, point * convex, CL &l, point &p){
ab11   if (line_throw_convex(n, convex, l, p)) {
ff30     if (dmult(p, l.b, l.a) >= -eps) return true;return false;
95cf   }
438e   return false;
95cf }
427e
427e // 求凸包直径, 输入要求顺时针输入凸包，没有共线的点
9c74 double convex_diameter(int n, point *con) {
571d   int q=1;double ans=0;
9c16   for(int p=0;p<n;++p)
4506   {
f1dc     while(xmult(con[(p+1)%n],con[(q+1)%n],con[p])<xmult(con[(p+1)%n],con[q],con[p]))
78df       q=(q+1)%n;
605a     ans=max(ans,max(dis(con[p],con[q]),dis(con[(p+1)%n],con[(q+1)%n])));
95cf   }
4206   return ans;
95cf }
427e
427e // 求凸包最小截面，输入要求顺时针输入凸包
2bc8 double convex_min_section(int n, point * con){
987e   int q=1;double ans=1000000000;
9c16   for(int p=0;p<n;++p)
4506   {
f1dc     while(xmult(con[(p+1)%n],con[(q+1)%n],con[p])<xmult(con[(p+1)%n],con[q],con[p]))
78df       q=(q+1)%n;
adf5     ans=min(ans,disptoline(con[q], con[p], con[p + 1]));
95cf   }
4206   return ans;
95cf }
ff80 double convex_min_section2(int n, point * con){
c166   double l1 = 1000000000;
6c2f   for (int i = 0; i < n; ++i){
3cec     point a = con[i] - con[(i + 1) % n];
5394     point b = vector_throw_convex(n, con, a);
746a     l1 = min(l1, disptoline(b, con[i], con[i + 1]));
95cf   }
66fb   return l1;
95cf }
427e
427e // 求两个不包含的凸包的最短距离, 逆时针输入背包
27ad double convex_min_dis(int n, point *a, int m, point *b){
4ab2   int p1 = 0, p2 = 0;
5db6   double ans = 1<<30;
85c3   for (int i = 0; i < n; ++i)
6f87     if (a[i].y < a[p1].y) p1 = i;
e725   for (int i = 0; i < m; ++i)
ca57     if (b[i].y > b[p2].y) p2 = i;
6c2f   for (int i = 0; i < n; ++i){
ff12     double t = xmult(b[(p2 + 1) % m], a[p1], a[(p1 + 1) % n]);
474d     t -= xmult(b[p2], a[p1], a[(p1 + 1) % n]);
427e     //cout<<p1<<' '<<p2<<' '<<t<<endl;
2c1f     if (_sign(t) == 1){
c706       ans = min(ans, disptoseg(a[p1], b[p2], b[(p2 + 1) % m]));
8d05       p2 = (p2 + 1) % m;
215f       --i;
09a7     }else if (_sign(t) == 2){
1c6e       ans = min(ans, disptoseg(b[p2], a[p1], a[(p1 + 1) % n]));
c583       p1 = (p1 + 1) % n;
8e2e     }else{
50f3       ans = min(ans, dissegtoseg(a[p1], a[(p1 + 1) % n], b[p2], b[(p2 + 1) % m]));
c583       p1 = (p1 + 1) % n;
8d05       p2 = (p2 + 1) % m;
95cf     }
95cf   }
4206   return ans;
95cf }
427e
427e // 求多边形中最长的线段，的长度，线段储存在 l 中
25a1 double inside_polygon_max(int n,point* p, line &l){
e5cd   double len = 0;
85c3   for (int i = 0; i < n; ++i)
dd26     for (int j = i + 1; j < n; ++j){
374e       vector<point> points;
5603       points.clear();
aaca       points.push_back(p[i]);
3329       points.push_back(p[j]);
1aaa       for (int a = 0; a < n; ++a)
231a         for (int b = a + 1; b < n; ++b){
```

```
dbc2           if (a == i) continue;
3695           if (parallel(p[i], p[j], p[a], p[b]))
b333             continue;
f9ae           point p1 = intersection(p[i], p[j], p[a], p[b]);
cb8b           if (dmult(p[a], p[b], p1) <= 0){
a62b             points.push_back(p1);
95cf           }
95cf         }
52c1       sort(points.begin(), points.end());
427e
bd55       int s = 0;
2927       for (int k = 0; k < points.size() - 1; ++k){
d563         if (zero(dis(points[k], points[k + 1]))) continue;
8ac5         point p1;
1667         p1 = (points[k] + points[k + 1]) / 2;
8984         if (inside_polygon(p1, n, p)) continue;
154a         double d = dis(points[s], points[k + 1]);
db17         if (len < d){
aaed           len = d;
bff1           l.a = points[s];
71e0           l.b = points[k + 1];
95cf         }
69cc         s = k + 1;
95cf       }
5be3       double d = dis(points[s], points[points.size() - 1]);
db17       if (len < d){
aaed         len = d;
bff1         l.a = points[s];
352a         l.b = points[points.size() - 1];
95cf       }
95cf     }
1891   return len;
95cf }
427e
427e // 判断点知否在半平面内，平面位于向量左侧
917a bool phplaneout(CL &l, CP &p){
45ae   return xmult(p, l.b, l.a) > eps;
95cf }
427e
427e // 求半平面交，平面位于向量左侧
e484 int halfpanelcross(int n, line *lines, point *p){
a0f7   int i;
85c3   for (int i = 0; i < n; ++i)
```

```
    lines[i].getang(); 4d87
  sort(lines, lines + n); efa0
  int m = 1; dcba
  for (int i = 1; i < n; ++i) bc29
    if (!zero(lines[i].ang - lines[i - 1].ang)) fa79
      lines[m++] = lines[i]; aa88
  n = m; 7852
  int bot = 0, top = 1; dad4
  for (int i = 2; i < n; ++i){ a007
    if ((parallel(lines[top], lines[top - 1])||parallel(lines[bot], lines[bot + 1]))) f9fe
      return 0; 7021
    while ((bot < top)&&(xmult(intersection(lines[top], lines[top - 1]), lines[i].b, lines[i].a) > eps)) ba77
      --top; f959
    while ((bot < top)&&(xmult(intersection(lines[bot], lines[bot + 1]), lines[i].b, lines[i].a) > eps)) c049
      ++bot; 5425
    ++top; c56b
    lines[top] = lines[i]; 973c
  } 95cf
  while ((bot < top)&&(xmult(intersection(lines[top], lines[top - 1]), lines[bot].b, lines[bot].a) > eps)) f6dc
    --top; f959
  while ((bot < top)&&(xmult(intersection(lines[bot], lines[bot + 1]), lines[top].b, lines[top].a) > eps)) 1b45
    ++bot; 5425
  if (top <= bot + 1) return 0; 580b
  n = 0; 91c9
  for (int i = bot; i < top; ++i) 9c73
    p[n++] = intersection(lines[i], lines[i + 1]); dfe9
  if (bot < top + 1) a1f8
    p[n++] = intersection(lines[bot], lines[top]); f5bc
  return n; c757
} 95cf
 427e
// 算法顺时针构造包含所有共线点的凸包graham,O(nlogn) 427e
point p1,p2; cc70
int graham_cp(const void* a,const void* b){ 91cb
  double ret=xmult(*((point*)a),*((point*)b),p1); 43a2
```

```
6307    return zero(ret)?(xmult(*((point*)a)
          ,*((point*)b),p2)>0?1:-1):(ret
          >0?1:-1);
95cf  }
a709  void _graham(int n,point* p,int& s,point*
         ch){
6e7d    int i,k=0;
396d    for (p1=p2=p[0],i=1;i<n;p2.x+=p[i].x,p2
          .y+=p[i].y,i++)
b2ba      if (p1.y-p[i].y>eps||(zero(p1.y-p[i].
            y)&&p1.x>p[i].x))
6d56        p1=p[k=i];
6584    p2.x/=n,p2.y/=n;
fc86    p[k]=p[0],p[0]=p1;
427e    //cout<<n<<endl;
725f    qsort(p+1,n-1,sizeof(point),graham_cp);
427e    //cout<<n<<endl;
501c    for (ch[0]=p[0],ch[1]=p[1],ch[2]=p[2],s
          =i=3;i<n;ch[s++]=p[i++])
1eaf      for (;s>2&&xmult(ch[s-2],p[i],ch[s
            -1])<-eps;s--);
95cf  }
427e
427e  // 构造凸包接口函数, 传入原始点集大小n, 点
        集p(p 原有顺序被打乱!)
427e  // 返回凸包大小, 凸包的点在convex 中
427e  // 参数maxsize 为1 包含共线点, 为0 不包含共
        线点, 缺省为1
427e  // 参数clockwise 为1 顺时针构造, 为0 逆时针
        构造, 缺省为1
427e  // 在输入仅有若干共线点时算法不稳定, 可能有
        此类情况请另行处理!
427e  // 不能去掉点集中重合的点
046c  int graham(int n,point* p,point* convex,
        int maxsize=1,int dir=1){
b7cd    point* temp=new point[n];
dee3    int s,i;
7ec2    _graham(n,p,s,temp);
79fb    for (convex[0]=temp[0],n=1,i=(dir?1:(s
          -1));dir?(i<s):i;i+=(dir?1:-1)){
cecf      if (maxsize||!zero(xmult(temp[i-1],
            temp[i],temp[(i+1)%s])))
02e5        convex[n++]=temp[i];
95cf      }
8562    delete []temp;
c757    return n;
95cf  }求入射边
427e
427e  //关于边的折射角，不考虑全反射，折射率为uvr
0ab4  line refraction(CL &u, CL &v, CP &p,
        double r){
57f5    line v2;
57f9    v2.a = p;
```

```
  v2.b.x = v.b.y - v.a.y + p.x;                      6bd4
  v2.b.y = v.a.x - v.b.x + p.y;                      99c4
  if (dmult(v2.b - v2.a, u.b - u.a) < 0)             efc1
    swap(v2.a, v2.b);
  double alpha = xmult(v2.b - v2.a, u.b -            10cd
     u.a) / len(v2.b - v2.a) / len(u.b -
    u.a);
  alpha = asin(alpha / r) + atan2(v2.b.y             d9eb
    - v2.a.y, v2.b.x - v2.a.x);
  v2.a = p;                                          57f9
  v2.b.x = 10 * cos(alpha) + v2.a.x;                 2a97
  v2.b.y = 10 * sin(alpha) + v2.a.y;                 33d7
  return v2;                                         9d48
}求入射边                                             95cf
                                                     427e
//关于凸包折射两次的情况，出射角保存在中，折             427e
  射率为vvr如果不相交，返回
//，false 不考虑镜面反射和入射到凸包角上的情           427e
  况
bool refreaction(int n, point p[], line &            0c20
  v, double r){
  int index = -1;                                    4e58
  line l[n];                                         c9aa
  point p1;                                          8ac5
  for (int i = 0; i < n; ++i){                       6c2f
    l[i].a = p[i];                                   bd2f
    l[i].b = p[(i + 1) % n];                         ab75
  }                                                  95cf
  for (int i = 0; i < n; ++i){                       6c2f
    if (parallel(v, l[i])) continue;                 8cf8
    point p2 = intersection(v, l[i]);                fada
    if (dmult(l[i].a, l[i].b, p2) >= 0)              a36f
      continue;
    if (dmult(p2, v.b, v.a) <= 0)                    6446
      continue;
    if ((index == -1)||(dis2(p2, v.a) <              fe41
      dis2(p1, v.a))){
      index = i;                                     8d74
      p1 = p2;                                       3524
    }                                                95cf
  }                                                  95cf
  if (index == -1) return 0;                         38b6
  swap(l[0], l[index]);                              2817
  v = refraction(v, l[0], p1, r);                    5df5
  index = -1;                                        17c3
  for (int i = 1; i < n; ++i){                       2892
    if (parallel(v, l[i])) continue;                 8cf8
    point p2 = intersection(v, l[i]);                fada
    if (dmult(l[i].a, l[i].b, p2) >= 0)              a36f
      continue;
    if (dmult(p2, v.b, v.a) <= 0)                    6446
      continue;
    if ((index == -1)||(dis2(p2, v.a) <              fe41
            dis2(p1, v.a))){
8d74        index = i;
3524        p1 = p2;
95cf      }
95cf    }
087b    swap(l[1], l[index]);
01a2    v = refraction(v, l[1], p1, 1 / r);
3361    return true;
95cf  }
```

2.6 面积

```
427e  // 计算三角形面积，输入三顶点
ab5f  double area_triangle(CP &p1,CP &p2,CP &p3
        ){
f8a3    return fabs(xmult(p1,p2,p3))/2;
95cf  }
243e  double area_triangle(double x1,double y1,
        double x2,double y2,double x3,double y3
        ){
1953    return fabs(xmult(x1,y1,x2,y2,x3,y3))
          /2;
95cf  }
427e
427e  // 计算三角形面积，输入三边长
7fb2  double area_triangle(double a,double b,
        double c){
f36a    double s=(a+b+c)/2;
7a86    return sqrt(s*(s-a)*(s-b)*(s-c));
95cf  }
427e
427e  // 计算多边形面积，顶点按顺时针或逆时针给出
427e  // 顺时针的时候，面积为负，逆时针的时候面积为
        正
2ed4  double area_polygon(int n,point* p){
0934    double s1=0,s2=0;
a0f7    int i;
2dbf    for (i=0;i<n;i++)
e3ae      s1+=p[(i+1)%n].y*p[i].x,s2+=p[(i+1)%n
          ].y*p[(i+2)%n].x;
1109    return (s1-s2)/2;
95cf  }
```

2.7 球面

```
427e  // 计算圆心角lat 表示纬度,-90<=w<=90,lng 表
        示经度
427e  // 返回两点所在大圆劣弧对应圆心
        角,0<=angle<=pi
8176  double angle(double lng1,double lat1,
        double lng2,double lat2){
533a    double dlng=fabs(lng1-lng2)*pi/180;
88a6    while (dlng>=pi+pi)
a612      dlng-=pi+pi;
  if (dlng>pi)                                d991
    dlng=pi+pi-dlng;                          ea50
  lat1*=pi/180,lat2*=pi/180;                  6a31
  return acos(cos(lat1)*cos(lat2)*cos(        dc31
    dlng)+sin(lat1)*sin(lat2));
}                                             95cf
                                              427e
// 计算距离,r 为球半径                          427e
double line_dist(double r,double lng1,        3a23
  double lat1,double lng2,double lat2){
  double dlng=fabs(lng1-lng2)*pi/180;         533a
  while (dlng>=pi+pi)                         88a6
    dlng-=pi+pi;                              a612
  if (dlng>pi)                                d991
    dlng=pi+pi-dlng;                          ea50
  lat1*=pi/180,lat2*=pi/180;                  6a31
  return r*sqrt(2-2*(cos(lat1)*cos(lat2)*     24df
    cos(dlng)+sin(lat1)*sin(lat2)));
}                                             95cf
                                              427e
// 计算球面距离,r 为球半径                      427e
inline double sphere_dist(double r,double     1bec
   lng1,double lat1,double lng2,double
  lat2){
  return r*angle(lng1,lat1,lng2,lat2);        5db0
}                                             95cf
```

2.8 圆

```
// 判直线和圆相交，包括相切                      427e
int intersect_line_circle(CP &c,double r,     128f
  CP &l1,CP &l2){
  return disptoline(c,l1,l2)<r+eps;           c721
}                                             95cf
int intersect_line_circle(CC &c,CP &l1,CP     641a
   &l2){
  return disptoline(c.c,l1,l2)<c.r+eps;       e25b
}                                             95cf
                                              427e
// 判线段和圆相交，包括端点和相切                427e
int intersect_seg_circle(CP &c,double r,      a4a8
  CP &l1,CP &l2){
  double t1=dis(c,l1)-r,t2=dis(c,l2)-r;       7ace
  point t=c;                                  481d
  if (t1<eps||t2<eps)                         6bfd
    return t1>-eps||t2>-eps;                  b703
  t.x+=l1.y-l2.y;                             524a
  t.y+=l2.x-l1.x;                             9773
  return xmult(l1,c,t)*xmult(l2,c,t)<eps      7706
    &&disptoline(c,l1,l2)-r<eps;
}                                             95cf
int intersect_seg_circle(CC &c,CP &l1,CP      e3cf
  &l2){
  double t1=dis(c.c,l1)-c.r,t2=dis(c.c,l2     63c9
         )-c.r;
3bc7   point t=c.c;
6bfd   if (t1<eps||t2<eps)
b703     return t1>-eps||t2>-eps;
524a   t.x+=l1.y-l2.y;
9773   t.y+=l2.x-l1.x;
78b0   return xmult(l1,c.c,t)*xmult(l2,c.c,t)<
         eps&&disptoline(c.c,l1,l2)-c.r<eps;
95cf }
427e // 判圆和圆相交, 包括相切
e4fa int intersect_circle_circle(CP &c1,double
        r1,CP &c2,double r2){
9676   return dis(c1,c2)<r1+r2+eps&&dis(c1,c2)
         >fabs(r1-r2)-eps;
95cf }
da9a int intersect_circle_circle(CC &c1,CC &c2
       ){
14e2   return dis(c1.c,c2.c)<c1.r+c2.r+eps&&
         dis(c1.c,c2.c)>fabs(c1.r-c2.r)-eps;
95cf }
427e
427e // 计算圆上到点p 最近点, 如p 与圆心重合, 返
       回p 本身
f391 point dot_to_circle(CP &c,double r,CP &p)
       {
03f5   point u,v;
82e2   if (dis(p,c)<eps)
e149     return p;
6b5b   u.x=c.x+r*fabs(c.x-p.x)/dis(c,p);
e0a3   u.y=c.y+r*fabs(c.y-p.y)/dis(c,p)*((c.x-
         p.x)*(c.y-p.y)<0?-1:1);
8ae0   v.x=c.x-r*fabs(c.x-p.x)/dis(c,p);
81cd   v.y=c.y-r*fabs(c.y-p.y)/dis(c,p)*((c.x-
         p.x)*(c.y-p.y)<0?-1:1);
ceec   return dis(u,p)<dis(v,p)?u:v;
95cf }
521e point dot_to_circle(CC &c,CP &p){
03f5   point u,v;
3905   if (dis(p,c.c)<eps)
e149     return p;
525f   u.x=c.c.x+c.r*fabs(c.c.x-p.x)/dis(c.c,p
         );
0882   u.y=c.c.y+c.r*fabs(c.c.y-p.y)/dis(c.c,p
         )*((c.c.x-p.x)*(c.c.y-p.y)<0?-1:1);
2d9f   v.x=c.c.x-c.r*fabs(c.c.x-p.x)/dis(c.c,p
         );
fc50   v.y=c.c.y-c.r*fabs(c.c.y-p.y)/dis(c.c,p
         )*((c.c.x-p.x)*(c.c.y-p.y)<0?-1:1);
ceec   return dis(u,p)<dis(v,p)?u:v;
95cf }
427e
427e // 计算直线与圆的交点, 保证直线与圆有交点
427e // 计算线段与圆的交点可用这个函数后判点是否在
```

```
  线段上
void intersection_line_circle(CP &c,           c8d1
  double r,CP &l1,CP &l2,point& p1,point&
   p2){
  point p=c;                                    e36e
  double t;                                     3337
  p.x+=l1.y-l2.y;                               e339
  p.y+=l2.x-l1.x;                               4399
  p=intersection(p,c,l1,l2);                    1b68
  t=sqrt(r*r-dis(p,c)*dis(p,c))/dis(l1,l2       d753
    );
  p1.x=p.x+(l2.x-l1.x)*t;                       f468
  p1.y=p.y+(l2.y-l1.y)*t;                       7618
  p2.x=p.x-(l2.x-l1.x)*t;                       d8a7
  p2.y=p.y-(l2.y-l1.y)*t;                       24fc
}                                               95cf
void intersection_line_circle(CC &c,CP &        c26e
  l1,CP &l2,point& p1,point& p2){
  point p=c.c;                                  92cd
  double t;                                     3337
  p.x+=l1.y-l2.y;                               e339
  p.y+=l2.x-l1.x;                               4399
  p=intersection(p,c.c,l1,l2);                  c608
  t=sqrt(c.r*c.r-dis(p,c.c)*dis(p,c.c))/        3855
    dis(l1,l2);
  p1.x=p.x+(l2.x-l1.x)*t;                       f468
  p1.y=p.y+(l2.y-l1.y)*t;                       7618
  p2.x=p.x-(l2.x-l1.x)*t;                       d8a7
  p2.y=p.y-(l2.y-l1.y)*t;                       24fc
}                                               95cf
                                                427e
// 计算圆与圆的交点, 保证圆与圆有交点圆心不       427e
  重合,
void intersection_circle_circle(CP &c1,         0999
  double r1,CP &c2,double r2,point& p1,
  point& p2){
  point u,v;                                    03f5
  double t;                                     3337
  t=(1+(r1*r1-r2*r2)/dis(c1,c2)/dis(c1,c2       b814
    ))/2;
  u.x=c1.x+(c2.x-c1.x)*t;                       fcc3
  u.y=c1.y+(c2.y-c1.y)*t;                       3c66
  v.x=u.x+c1.y-c2.y;                            20b2
  v.y=u.y-c1.x+c2.x;                            c44e
  intersection_line_circle(c1,r1,u,v,p1,        bb81
    p2);
}                                               95cf
void intersection_circle_circle(CC &c1,CC       8068
   &c2,point& p1,point& p2){
  point u,v;                                    03f5
  double t;                                     3337
  t=(1+(c1.r*c1.r-c2.r*c2.r)/dis(c1.c,c2.       ad39
    c)/dis(c1.c,c2.c))/2;
```

```
8414   u.x=c1.c.x+(c2.c.x-c1.c.x)*t;
b2d6   u.y=c1.c.y+(c2.c.y-c1.c.y)*t;
4218   v.x=u.x+c1.c.y-c2.c.y;
4b8b   v.y=u.y-c1.c.x+c2.c.x;
30f9   intersection_line_circle(c1.c,c1.r,u,v,p1,p2);
95cf }
427e
427e // 判断圆在多边形内, 顶点按顺时针或逆时针给出,offset 为多边形坐标上限
9451 bool inside_circle_polygon(CP &c, double r, int n, point * polygon){
d1de   if (!inside_polygon(c, n, polygon, 1))
438e     return false;
85c3   for (int i = 0; i < n; ++i)
b348     if (disptoline(c,polygon[i], polygon[(i + 1) % n]) < r)
438e       return false;
3361   return true;
95cf }
0dc3 bool inside_circle_polygon(CC &c, int n, point * polygon){
a224   if (!inside_polygon(c.c, n, polygon, 1))
438e     return false;
85c3   for (int i = 0; i < n; ++i)
a14b     if (disptoline(c.c,polygon[i], polygon[(i + 1) % n]) < c.r)
438e       return false;
3361   return true;
95cf }
427e
427e // 判断多边形在圆内，包括圆上
5c07 bool inside_polygon_circle(CP &c, double r, int n, point *polygon){
85c3   for (int i = 0; i < n; ++i)
1b91     if (dis2(c, polygon[i]) >= r * r)
438e       return false;
3361   return true;
95cf }
6a80 bool inside_polygon_circle(CC &c, int n, point *polygon){
85c3   for (int i = 0; i < n; ++i)
5c9f     if (dis2(c.c, polygon[i]) >= c.r * c.r)
438e       return false;
3361   return true;
95cf }
427e
427e // 求圆外一点与圆的切线, 返回两个切点
218e void tangent_point_circle(CP &c, double r, CP &p, point &a, point &b){
00e3   double d = dis(c, p);
```

```
  double angp = acos(r / d);                                  0e20
  double ango = atan2(p.y - c.y, p.x - c.x);                  736d
  a.x = c.x + r * cos(ango + angp);                           0150
  a.y = c.y + r * sin(ango + angp);                           fc8b
  b.x = c.x + r * cos(ango - angp);                           8b80
  b.y = c.y + r * sin(ango - angp);                           0c1e
}                                                             95cf
void tangent_point_circle(CC &c, CP &p, point &a, point &b){  6800
  double d = dis(c.c, p);                                     ad5f
  double angp = acos(c.r / d);                                5c2a
  double ango = atan2(p.y - c.c.y, p.x - c.c.x);              5922
  a.x = c.c.x + c.r * cos(ango + angp);                       e086
  a.y = c.c.y + c.r * sin(ango + angp);                       a8d8
  b.x = c.c.x + c.r * cos(ango - angp);                       b6f5
  b.y = c.c.y + c.r * sin(ango - angp);                       aa9f
}                                                             95cf
                                                              427e
// 求内切圆，返回两个切线                                         427e
void incut_circle_circle(CP &c1,double r1,CP &c2,double r2,line& l1,line& l2){  0d4d
  double d = sqrt(dis2(c1, c2) - sqr(r1 + r2));               b709
  point p1, p2;                                               cc70
  intersection_circle_circle(c1, r1 + r2, c2, d, p1, p2);     d070
  l1.a = (p1 * r1 + c1 * r2) / (r1 + r2);                     6042
  l1.b = l1.a + (c2 - p1);                                    6bb3
  l2.a = (p2 * r1 + c1 * r2) / (r1 + r2);                     4a4c
  l2.b = l2.a + (c2 - p2);                                    ea2d
}求原点原语扇形的夹角                                           95cf
                                                              427e
//                                                            427e
double area_circle_angle(CP &p1, CP &p2, CP &c, double r){    5ad0
  double alpha = fabs(atan2(p1.y - c.y, p1.x - c.x) - atan2(p2.y - c.y, p2.x - c.x));  db38
  if (alpha > pi) alpha = 2 * pi - alpha;                     9617
  return alpha / 2 * r * r;                                   25ea
}                                                             95cf
                                                              427e
// 求三角形的外接圆                                              427e
void  circleoftri(CP &a, CP &b, CP &c, circle &tmp){          6ff9
  tmp.c = circumcenter(a,b,c);                                d028
  tmp.r = dis(a, tmp.c);                                      cdd7
}                                                             95cf
                                                              427e
// 求包含n 个给定点的的最小圆， n <= 3                             427e
void min_circle_reduce(int n, point *p,                       8504
      circle &tmp){
427e    //cout<<n<<endl;
7707    if (n == 0) tmp.r = -2;
a7ed    else if (n == 1){
c330      tmp.c = p[0];
0541      tmp.r = 0;
02a8    }else if (n == 2){
fa16      tmp.r = dis(p[0], p[1]) / 2;
8222      tmp.c = (p[0] + p[1]) / 2;
119b    }else if (n == 3)
87a4      circleoftri(p[0], p[1], p[2], tmp);
95cf  }
427e
e0ae  void min_circle(int n, point *p, int m,
        point *down, circle &c){
427e
425e    min_circle_reduce(m, down, c);
1ae8    if (m == 3) return;
6c2f    for (int i = 0; i < n; ++i){
427e      //cout<<i<<"  "<<n<<endl;
427e      //cout<<dis(p[i], c.c)<<' '<<c.r<<
            endl;
899e      if (dis(p[i], c.c) > c.r){
427e        //cout<<m<<"yes"<<endl;
03c5        down[m] = p[i];
85d9        min_circle(i, p, m + 1, down, c);
fe45        point tmp = p[i];
f1bb        for (int j = i; j >= 1; --j)
a042          p[j] = p[j - 1];
9161        p[0] = tmp;
95cf      }
95cf    }
95cf  }
427e
427e  // 求包含n 个给定点的最小圆
71a8  void min_circle(int n, point *p, circle &
        c){
ab81    point down[3];
9fff    min_circle(n, p, 0,down,c);
95cf  }求圆和三角形
427e
427e  //{c p1 p2的相交面积}
4c91  double area_triangle_circle(CP &c, double
         r, CP &p1, CP &p2){
8059    double x = xmult(p2, c, p1);
3230    int flag = ((x)>eps?1:((x)<-eps?-1:0));
1be0    if (flag == 0) return 0;
c3b3    double r2 = sqr(r);
6c66    double s = 0, l1 = dis2(p1, c), l2 =
          dis2(p2, c);
64ce    if ((l1 <= r2)&&(l2 <= r2))
f56c      return area_triangle(p2, c, p1) *
            flag;
```

```
  if ((l1 > r2)&&(l2 > r2)){                  1e19
    point p3, p4;                             98ff
    s = area_circle_angle(p2, p1, c, r);      fbb4
    if (disptoseg(c, p1, p2) < r){            a4ed
      intersection_line_circle(c, r, p1,      4917
        p2, p3, p4);
      if (dis2(p3, p1) > dis2(p4, p1))        7d00
        swap(p3, p4);
      s -= area_circle_angle(p3, p4, c, r     a289
        ) - area_triangle(p3, c, p4);
      }                                       95cf
    return s * flag;                          c890
  }                                           95cf
  if (l1 < l2){                               bcd6
    point p3, p4;                             98ff
    intersection_line_circle(c, r, p1, p2     4917
      , p3, p4);
    if (dmult(p3, p2, p1) <= 0) p3 = p4;      912e
    s = area_triangle(p1, p3, c) +            f0b8
      area_circle_angle(p3, p2, c, r);
    return s * flag;                          c890
  }else{                                      8e2e
    point p3, p4;                             98ff
    intersection_line_circle(c, r, p1, p2     4917
      , p3, p4);
    if (dmult(p3, p1, p2) <= 0) p3 = p4;      f453
    s = area_triangle(p2, p3, c) +            8ee3
      area_circle_angle(p3, p1, c, r);
    return s * flag;                          c890
  }                                           95cf
}求圆和多边形的相交面积                           95cf
                                              427e
//                                            427e
double area_polygon_circle(int n, point p     baa7
  [], CP &c, double r){
  double ans = 0;                             753f
  for (int i = 0; i < n; ++i)                 85c3
    ans += area_triangle_circle(c, r, p[i     3b9b
      ], p[(i + 1) % n]);
  return fabs(ans);                           80ec
}                                             95cf
```

2.9 网格

```
#define abs(x) ((x)>0?(x):-(x))               058c
struct point{int x,y;};                       29c0
                                              427e
int gcd(int a,int b){                         0d5b
  return b?gcd(b,a%b):a;                      5fd6
}                                             95cf
                                              427e
// 多边形上的网格点个数                           427e
int grid_onedge(int n,point* p){              bd90
  int i,ret=0;                                54c1
2dbf    for (i=0;i<n;i++)
b3cc      ret+=gcd(abs(p[i].x-p[(i+1)%n].x),abs
            (p[i].y-p[(i+1)%n].y));
ee0f    return ret;
95cf  }
427e
427e  // 多边形内的网格点个数
b273  int grid_inside(int n,point* p){
54c1    int i,ret=0;
2dbf    for (i=0;i<n;i++)
0639      ret+=p[(i+1)%n].y*(p[i].x-p[(i+2)%n].
            x);
0ba0    return (abs(ret)-grid_onedge(n,p))/2+1;
95cf  }
```

2.10 区域中点集个数

```
427e  // 求p1 中任意三点切割的七个区域的p1 点集的
        个数, 要求三点不共线
427e  // 求xy 到xz 角度里点的个数
cfb5  int pointinang(int x, int y, int z, int m
        , int f[][MAXN], int index[][MAXN]){
3c28    if (index[x][z] < index[x][y]) return m
          + f[x][z] - f[x][y] + 1;
03df    return f[x][z] - f[x][y] - 1;
95cf  }
427e
427e  // 求三角形xyz 中的点的个数, 输入总的点的个
        数,
427e  //f[x][y] 表示x 为中心ang 比y 小的点的个数
427e  //h[x][y] 表示xy 左边的点的个数
427e  //index[x][y] 表示x 为中心极角排序后的序
e42a  int pointintri(point p[], int x, int y,
        int z, int m, int f[][MAXN], int h[][
        MAXN], int index[][MAXN]){
283b    if (xmult(p[z], p[y], p[x]) > 0) swap(y
          , z);
53ef    int a = h[x][z] + h[y][x] + h[z][y];
5a59    a += pointinang(x, y, z, m, f, index);
f5d7    a += pointinang(y, z, x, m, f, index);
150c    a += pointinang(z, x, y, m, f, index);
fd8e    a -= 2 * m;
5ffd    return a;
95cf  }
427e
427e  // 求p1 任意三点划分的区域的点的个数, 要求三
        点不共线
427e  //f[x][y] 表示x 为中心ang 比y 小的点的个数
427e  //h[x][y] 表示xy 左边的点的个数
427e  //index[x][y] 表示x 为中心极角排序后的序
614b  void pointinarea(int n, point p1[], int f
        [][MAXN], int h[][MAXN], int index[][
        MAXN]){
60da    point p[2 * MAXN];
```

```
  for (int i = 0; i < n; ++i){                 6c2f
    int cnt = 0;                               8abb
    for (int j = 0; j < n; ++j)                fde8
      if (i != j){                             4a23
        p[cnt] = p1[j];                        ca58
        p[cnt].index = j;                      28c2
        p[cnt++].ang = atan2(p1[j].y - p1      2f66
          [i].y, p1[j].x - p1[i].x);
      }                                        95cf
    sort(p, p + cnt, cmp);                     b073
    for (int j = 0; j < cnt; ++j){             07de
      p[j + cnt] = p[j];                       fcfa
      p[j + cnt].ang += 2 * pi;                ceb9
    }                                          95cf
    for (int k = 0, j = 0, mine = 0, l =       07b3
      0; k < cnt; ++k){
      while (p[j].ang - p[k].ang < pi){        9b9c
        ++mine;                                ae6d
        ++j;                                   917f
      }                                        95cf
      --mine;                                  5021
      h[i][p[k].index] = mine;                 ca69
      f[i][p[k].index] = l;                    c92b
      index[i][p[k].index] = k;                e5d0
      ++l;                                     713f
    }                                          95cf
  }                                            95cf
}                                              95cf
                                               427e
                                               427e
// 求p1 中任意三点切割的七个区域的p2 点集的     427e
  个数, 要求三点不共线
// 求xy 到xz 角度里点的个数                    427e
int pointinang2(int x, int y, int z, int       251e
  m, int f[][MAXN], int index[][MAXN]){
  if (index[x][z] < index[x][y]) return m      35b8
    + f[x][z] - f[x][y];
  return f[x][z] - f[x][y];                    bd1e
}                                              95cf
                                               427e
// 求三角形xyz 中的点的个数, 输入总的点的个     427e
  数,
//f[x][y] 表示x 为中心ang 比y 小的点的个数     427e
//h[x][y] 表示xy 左边的点的个数                427e
//index[x][y] 表示x 为中心极角排序后的序       427e
int pointintri2(point p[], int x, int y,       73ff
  int z, int m, int f[][MAXN], int h[][
  MAXN], int index[][MAXN]){
  if (xmult(p[z], p[y], p[x]) > 0) swap(y      283b
    , z);
  int a = h[x][z] + h[y][x] + h[z][y];         53ef
  a += pointinang2(x, y, z, m, f, index);      2851
  a += pointinang2(y, z, x, m, f, index);      61bc
```

```
164e    a += pointinang2(z, x, y, m, f, index);
fd8e    a -= 2 * m;
5ffd    return a;
95cf  }
427e
427e  // 求p1 任意三点划分的区域的p2 点的个数，要求三点不共线
427e  //f[x][y] 表示x 为中心ang 比y 小的点的个数
427e  //h[x][y] 表示xy 左边的点的个数
427e  //index[x][y] 表示x 为中心极角排序后的序
3cbd  void pointinarea2(int n, point p1[], int m, point p2[], int f[][MAXN], int h[][MAXN], int index[][MAXN]){
9941    point p[2 * (MAXN + MAXM)];
6c2f    for (int i = 0; i < n; ++i){
8abb      int cnt = 0;
fde8      for (int j = 0; j < n; ++j)
4a23        if (i != j){
ca58          p[cnt] = p1[j];
28c2          p[cnt].index = j;
2f66          p[cnt++].ang = atan2(p1[j].y - p1[i].y, p1[j].x - p1[i].x);
95cf        }
6613      for (int j = 0; j < m; ++j){
acb5          p[cnt] = p2[j];
6d16          p[cnt].index = n + j;
3745          p[cnt++].ang = atan2(p2[j].y - p1[i].y, p2[j].x - p1[i].x);
95cf        }
b073      sort(p, p + cnt, cmp);
07de      for (int j = 0; j < cnt; ++j){
fcfa        p[j + cnt] = p[j];
ceb9        p[j + cnt].ang += 2 * pi;
95cf      }
07b3      for (int k = 0, j = 0, mine = 0, l = 0; k < cnt; ++k){
9b9c        while (p[j].ang - p[k].ang < pi){
8ddd          if (p[j].index >= n) ++mine;
917f          ++j;
95cf        }
85b2        if (p[k].index < n){
ca69          h[i][p[k].index] = mine;
c92b          f[i][p[k].index] = l;
e5d0          index[i][p[k].index] = k;
95cf        }
2410        if (p[k].index >= n){
5021          --mine;
713f          ++l;
95cf        }
95cf      }
95cf    }
95cf  }
```

3 三维计算几何

3.1 定义

```
#define eps 1e-8                                      652e
#define fabs(x) ((x) > 0? (x): -(x))                  c1b0
#define zero(x) (fabs(x) < eps)                       0102
#define sqr(x) ((x)*(x))                              dca2
#define _sign(x) ((x)>eps?1:((x)<-eps?2:0))           12d8
const double pi = acos(-1);                           13f1
                                                      427e
// 点的定义                                            427e
struct point3{                                        b6b2
  double x, y, z;                                     9d7e
  point3(){x = 0;y = 0;z = 0;}                        6c19
  point3(double sx, double sy, double sz){            18ee
    x = sx;                                           e87b
    y = sy;                                           d22b
    z = sz;                                           826b
  }                                                   95cf
  bool operator <(const point3 &b)const{              548e
    if (b.x == x){                                    1737
      if (y == b.y) return z < b.z;                   e65c
      return y < b.y;                                 326b
    }                                                 95cf
    return x < b.x;                                   66d1
  }                                                   95cf
  point3 operator - (const point3 &b) const {         bb9f
    point3 a;                                         d6bc
    a.x = x - b.x;                                    d53d
    a.y = y - b.y;                                    5365
    a.z = z - b.z;                                    eb85
    return a;                                         5ffd
  }                                                   95cf
  point3 operator + (const point3 &b) const{          c055
    point3 a;                                         d6bc
    a.x = x + b.x;                                    7683
    a.y = y + b.y;                                    70a0
    a.z = z + b.z;                                    ee56
    return a;                                         5ffd
  }                                                   95cf
  point3 operator / (const double &c) const{          1731
    point3 a;                                         d6bc
    a.x = x / c;                                      225c
    a.y = y / c;                                      414d
    a.z = z / c;                                      155b
    return a;                                         5ffd
  }                                                   95cf
4f46    point3 operator * (const double &c) const{
d6bc      point3 a;
7aa6      a.x = x * c;
9a5c      a.y = y * c;
f068      a.z = z * c;
5ffd      return a;
95cf    }
412f    bool operator == (const point3 &p) const {
84cf      return zero(x - p.x)&&zero(y - p.y)&&zero(z - p.z);
95cf    }
c712    friend ostream& operator << (ostream &out, const point3 &a);
329b  };
1da1  ostream& operator << (ostream &out, const point3 &a){
5d5d    out<<a.x<<' '<<a.y<<' '<<a.z;
d324    return out;
95cf  }
427e
427e  // 边定义
3cfc  struct line3{
2f60    point3 a, b;
2f79    line3(){};
64c6    line3(const point3 &p1, const point3 &p2){
0fa8      a = p1;
ce41      b = p2;
95cf    }
e69c    friend ostream& operator << (ostream &out, const line3 &a);
329b  };
0e98  ostream& operator << (ostream &out, const line3 &a){
ad81    out<<a.a<<' '<<a.b<<' ';
d324    return out;
95cf  }
427e
427e  // 面定义
a1a2  struct plane3{point3 a,b,c;};
```

3.2 点线面

```
427e  // 计算cross product U x V
c4b1  point3 xmult(const point3 &u, const point3 &v){
92b5    point3 ret;
97a6    ret.x = u.y * v.z - v.y * u.z;
55e0    ret.y = u.z * v.x - u.x * v.z;
cf98    ret.z = u.x * v.y - u.y * v.x;
ee0f    return ret;
95cf  }
```

```
                                                          427e
// 计算dot product U . V                                  427e
double dmult(const point3 &u, const point3 &v){           f42e
  return u.x * v.x + u.y * v.y + u.z * v.z;               49f2
}                                                         95cf
                                                          427e
// 取平面法向量                                           427e
point3 pvec(const plane3 &s){                             388c
  return xmult(s.a - s.b, s.b - s.c);                     07d5
}                                                         95cf
point3 pvec(const point3 &s1, const point3 &s2, const point3 &s3){   afb0
  return xmult(s1 - s2, s2 - s3);                         fd86
}                                                         95cf
                                                          427e
// 两点距离，单参数取向量大小                             427e
double dis(const point3 &p1, const point3 &p2){           9e30
  return sqrt(sqr(p1.x - p2.x) + sqr(p1.y - p2.y) + sqr(p1.z - p2.z));   fafb
}                                                         95cf
double dis2(const point3 &p1, const point3 &p2){          be8b
  return sqr(p1.x - p2.x) + sqr(p1.y - p2.y) + sqr(p1.z - p2.z);   5cdd
}                                                         95cf
                                                          427e
// 向量大小                                               427e
double len(const point3 &p){                              81ce
  return sqrt(sqr(p.x) + sqr(p.y) + sqr(p.z));            2aa8
}                                                         95cf
                                                          427e
// 判三点共线                                             427e
int dots_inline(const point3 &p1, const point3 &p2,const point3 &p3){   d4bb
  return len(xmult(p1 - p2, p2 - p3)) < eps;              f368
}                                                         95cf
                                                          427e
// 判四点共面                                             427e
int dots_onplane(const point3 &a, const point3 &b, const point3 &c, const point3 &d){   2e86
  return zero(dmult(pvec(a, b, c), d - a));               03fd
}                                                         95cf
                                                          427e
// 判点是否在线段上，包括端点和共线                       427e
int dot_online_in(const point3 &p, const line3 &l){       e21b
5cd9    return zero(len(xmult(p - l.a, p - l.b)))&&
7c2a      (l.a.x - p.x) * (l.b.x - p.x) < eps&&
0643      (l.a.y - p.y) * (l.b.y - p.y) < eps&&
4305      (l.a.z - p.z) * (l.b.z - p.z) < eps;
95cf  }
ac26  int dot_online_in(const point3 &p, const point3 &l1, const point3 &l2){
2feb    return zero(len(xmult(p - l1,p - l2)))&&
0a0b      (l1.x - p.x) * (l2.x - p.x) < eps&&
6f88      (l1.y - p.y) * (l2.y - p.y) < eps&&
51be      (l1.z - p.z) * (l2.z - p.z) < eps;
95cf  }
427e
427e  // 判点是否在线段上，不包括端点
bd50  int dot_online_ex(const point3 &p, const line3 &l){
02c3    return dot_online_in(p, l) && (!(p == l.a))&&(!(p == l.b));
95cf  }
f8de  int dot_online_ex(const point3 &p, const point3 &l1, const point3 &l2){
35c7    return dot_online_in(p, l1, l2) && (!(p == l1))&&(!(p == l2));
95cf  }
427e
427e  // 判点是否在空间三角形上，包括边界，三点共线无意义
9b58  int dot_inplane_in(const point3 &p, const plane3 &s){
de8c    return zero(len(xmult(s.a - s.b, s.a - s.c)) - len(xmult(p - s.a, p - s.b))-
febf      len(xmult(p - s.b, p - s.c)) - len(xmult(p - s.c, p - s.a)));
95cf  }
e9f4  int dot_inplane_in(const point3 &p, const point3 &s1, const point3 &s2, const point3 &s3){
cc89    return zero(len(xmult(s1 - s2, s1 - s3)) - len(xmult(p - s1, p - s2))-
daa7      len(xmult(p - s2, p - s3)) - len(xmult(p - s3, p - s1)));
95cf  }
427e
427e  // 判点是否在空间三角形上，不包括边界，三点共线无意义
6d33  int dot_inplane_ex(const point3 &p, const plane3 &s){
ac70    return dot_inplane_in(p, s) && len(xmult(p - s.a, p - s.b)) > eps &&
1c47      len(xmult(p - s.b, p - s.c)) > eps &&
f1de      len(xmult(p - s.c, p - s.a)) > eps;
95cf  }
bcde  int dot_inplane_ex(const point3 &p, const point3 &s1, const point3 &s2, const point3 &s3){
3430    return dot_inplane_in(p, s1, s2, s3) && len(xmult(p - s1, p - s2)) > eps &&
505b      len(xmult(p - s2, p - s3)) > eps &&
f5c1      len(xmult(p - s3, p - s1)) > eps;
95cf  }
427e
427e  // 判两点在线段同侧，点在线段上返回0，不共面无意义
7ef7  int same_side(const point3 &p1, const point3 &p2, const line3 &l){
d86a    return dmult(xmult(l.a - l.b, p1 - l.b), xmult(l.a - l.b, p2 - l.b)) > eps;
95cf  }
0410  int same_side(const point3 &p1, const point3 &p2,const point3 &l1, const point3 &l2){
0b72    return dmult(xmult(l1 - l2, p1 - l2), xmult(l1 - l2, p2 - l2)) > eps;
95cf  }
427e
427e  // 判两点在线段异侧，点在线段上返回0，不共面无意义
9477  int opposite_side(const point3 &p1, const point3 &p2, const line3 &l){
faa2    return dmult(xmult(l.a - l.b, p1 - l.b), xmult(l.a - l.b, p2 - l.b)) < -eps;
95cf  }
76a6  int opposite_side(const point3 &p1, const point3 &p2,const point3 &l1, const point3 &l2){
4ffb    return dmult(xmult(l1 - l2, p1 - l2), xmult(l1 - l2, p2 - l2)) < -eps;
95cf  }
427e
427e  // 判两点在平面同侧，点在平面上返回0
3b4b  int same_side(const point3 &p1, const point3 &p2, const plane3 &s){
e2bb    return dmult(pvec(s), p1 - s.a) * dmult(pvec(s), p2 - s.a) > eps;
95cf  }
7b77  int same_side(const point3 &p1, const point3 &p2, const point3 &s1, const point3 &s2, const point3 &s3){
94f6    return dmult(pvec(s1,s2,s3), p1 - s1) * dmult(pvec(s1,s2,s3), p2 - s1) > eps;
95cf  }
427e
427e  // 判两点在平面异侧，点在平面上返回0
```

```
9512 int opposite_side(const point3 &p1, const point3 &p2, const plane3 &s){
fe1f   return dmult(pvec(s), p1 - s.a) * dmult(pvec(s), p2 - s.a) < -eps;
95cf }
05f5 int opposite_side(const point3 &p1, const point3 &p2, const point3 &s1, const point3 &s2, const point3 &s3){
9067   return dmult(pvec(s1,s2,s3), p1 - s1) * dmult(pvec(s1,s2,s3), p2 - s1) < -eps;
95cf }
427e
427e // 判两直线平行
490a int parallel(const line3 &u, const line3 &v){
1ac2   return len(xmult(u.a - u.b, v.a - v.b)) < eps;
95cf }
7409 int parallel(const point3 &u1, const point3 &u2, const point3 &v1, const point3 &v2){
8751   return len(xmult(u1 - u2, v1 - v2)) < eps;
95cf }
427e
427e // 判两平面平行
3062 int parallel(const plane3 &u, const plane3 &v){
c6ae   return len(xmult(pvec(u) , pvec(v))) < eps;
95cf }
7d09 int parallel(const point3 &u1, const point3 &u2, const point3 &u3, const point3 &v1,point3 v2,point3 v3){
ef70   return len(xmult(pvec(u1, u2, u3), pvec(v1, v2, v3))) < eps;
95cf }
427e
427e // 判直线与平面平行
454a int parallel(const line3 &l, const plane3 &s){
31d4   return zero(dmult(l.a - l.b, pvec(s)));
95cf }
06d0 int parallel(const point3 &l1, const point3 &l2, const point3 &s1, const point3 &s2, const point3 &s3){
d311   return zero(dmult(l1 - l2, pvec(s1, s2, s3)));
95cf }
427e
427e // 判两直线垂直
b98a int perpendicular(const line3 &u, const line3 &v){
  return zero(dmult(u.a - u.b, v.a - v.b)); 85bf
} 95cf
int perpendicular(const point3 &u1, const point3 &u2, const point3 &v1, const point3 &v2){ b77b
  return zero(dmult(u1 - u2, v1 - v2)); 4547
} 95cf
 427e
// 判两平面垂直 427e
int perpendicular(const plane3 &u, const plane3 &v){ 0e75
  return zero(dmult(pvec(u), pvec(v))); da40
} 95cf
int perpendicular(const point3 &u1, const point3 &u2, const point3 &u3, const point3 &v1, const point3 &v2, const point3 &v3){ 75f2
  return zero(dmult(pvec(u1, u2, u3), pvec(v1, v2, v3))); 8919
} 95cf
 427e
// 判直线与平面平行 427e
int perpendicular(const line3 &l, const plane3 &s){ 16f9
  return len(xmult(l.a - l.b, pvec(s))) < eps; ff31
} 95cf
int perpendicular(const point3 &l1, const point3 &l2, const point3 &s1, const point3 &s2, const point3 &s3){ d6d3
  return len(xmult(l1 - l2, pvec(s1, s2, s3))) < eps; d501
} 95cf
 427e
// 判两线段相交，包括端点和部分重合 427e
int intersect_in(const line3 &u, const line3 &v){ dfe1
  if (!dots_onplane(u.a, u.b, v.a, v.b)) 5163
    return 0; 7021
  if (!dots_inline(u.a,u.b,v.a)||!dots_inline(u.a,u.b,v.b)) e887
    return !same_side(u.a,u.b,v)&&!same_side(v.a,v.b,u); d666
  return dot_online_in(u.a,v)||dot_online_in(u.b,v)||dot_online_in(v.a,u)||dot_online_in(v.b,u); efeb
} 95cf
int intersect_in(const point3 &u1, const point3 &u2, const point3 &v1, const point3 &v2){ 2b77
  if (!dots_onplane(u1, u2, v1, v2)) cedb
7021      return 0;
8fcb    if (!dots_inline(u1,u2,v1)||!
          dots_inline(u1,u2,v2))
4b79      return !same_side(u1,u2,v1,v2)&&!
            same_side(v1,v2,u1,u2);
cc40    return dot_online_in(u1,v1,v2)||
          dot_online_in(u2,v1,v2)||
          dot_online_in(v1,u1,u2)||
          dot_online_in(v2,u1,u2);
95cf  }
427e
427e  // 判两线段相交，不包括端点和部分重合
ea94  int intersect_ex(const line3 &u, const
        line3 &v){
6c51    return dots_onplane(u.a,u.b,v.a,v.b)&&
          opposite_side(u.a,u.b,v)&&
          opposite_side(v.a,v.b,u);
95cf  }
aa4a  int intersect_ex(const point3 &u1, const
        point3 &u2, const point3 &v1, const
        point3 &v2){
6424    return dots_onplane(u1,u2,v1,v2)&&
          opposite_side(u1,u2,v1,v2)&&
          opposite_side(v1,v2,u1,u2);
95cf  }
427e
427e  // 判线段与空间三角形相交，包括交于边界
        和(部分)包含
4e75  int intersect_in(const line3 &l, const
        plane3 &s){
8378    return !same_side(l.a,l.b,s)&&!
          same_side(s.a,s.b,l.a,l.b,s.c)&&
153b      !same_side(s.b,s.c,l.a,l.b,s.a)&&!
            same_side(s.c,s.a,l.a,l.b,s.b);
95cf  }
e9ac  int intersect_in(const point3 &l1, const
        point3 &l2, const point3 &s1, const
        point3 &s2, const point3 &s3){
3c9d    return !same_side(l1,l2,s1,s2,s3)&&!
          same_side(s1,s2,l1,l2,s3)&&
89da      !same_side(s2,s3,l1,l2,s1)&&!
            same_side(s3,s1,l1,l2,s2);
95cf  }
427e
427e  // 判线段与空间三角形相交，不包括交于边界
        和(部分)包含
2571  int intersect_ex(const line3 &l, const
        plane3 &s){
02e8    return opposite_side(l.a,l.b,s)&&
          opposite_side(s.a,s.b,l.a,l.b,s.c)&&
f9f7      opposite_side(s.b,s.c,l.a,l.b,s.a)&&
            opposite_side(s.c,s.a,l.a,l.b,s.b);
95cf  }
```

```
int intersect_ex(const point3 &l1, const        ebd9
  point3 &l2, const point3 &s1, const
  point3 &s2, const point3 &s3){
  return opposite_side(l1,l2,s1,s2,s3)&&        3e15
    opposite_side(s1,s2,l1,l2,s3)&&
    opposite_side(s2,s3,l1,l2,s1)&&             ebb0
      opposite_side(s3,s1,l1,l2,s2);
}                                               95cf
                                                427e
// 计算两直线交点，注意事先判断直线是否共面      427e
  和平行！
// 线段交点请另外判线段相交同时还是要判断是否    427e
  平行(!)
point3 intersection(const line3 &u, const       58cf
   line3 &v){
  point3 ret=u.a;                               87cc
  double t=((u.a.x-v.a.x)*(v.a.y-v.b.y)-(       273a
    u.a.y-v.a.y)*(v.a.x-v.b.x))
      /((u.a.x-u.b.x)*(v.a.y-v.b.y)-(u.a.       9cb3
        y-u.b.y)*(v.a.x-v.b.x));
  ret.x+=(u.b.x-u.a.x)*t;                       1143
  ret.y+=(u.b.y-u.a.y)*t;                       12e9
  ret.z+=(u.b.z-u.a.z)*t;                       1037
  return ret;                                   ee0f
}                                               95cf
point3 intersection(const point3 &u1,           b3fa
  const point3 &u2, const point3 &v1,
  const point3 &v2){
  point3 ret=u1;                                1fef
  double t=((u1.x-v1.x)*(v1.y-v2.y)-(u1.y       a7db
    -v1.y)*(v1.x-v2.x))
      /((u1.x-u2.x)*(v1.y-v2.y)-(u1.y-u2.       16f0
        y)*(v1.x-v2.x));
  ret.x+=(u2.x-u1.x)*t;                         a1f8
  ret.y+=(u2.y-u1.y)*t;                         fa1b
  ret.z+=(u2.z-u1.z)*t;                         d408
  return ret;                                   ee0f
}                                               95cf
                                                427e
// 计算直线与平面交点，注意事先判断是否平        427e
  行，并保证三点不共线！
// 线段和空间三角形交点请另外判断                427e
point3 intersection(const line3 &l, const       1f8f
   plane3 &s){
  point3 ret=pvec(s);                           a582
  double t=(ret.x*(s.a.x-l.a.x)+ret.y*(s.       f84a
    a.y-l.a.y)+ret.z*(s.a.z-l.a.z))/
    (ret.x*(l.b.x-l.a.x)+ret.y*(l.b.y-l.a       7e59
      .y)+ret.z*(l.b.z-l.a.z));
  ret.x=l.a.x+(l.b.x-l.a.x)*t;                  6247
  ret.y=l.a.y+(l.b.y-l.a.y)*t;                  6934
  ret.z=l.a.z+(l.b.z-l.a.z)*t;                  706b
  return ret;                                   ee0f
95cf }
2ec4 point3 intersection(const point3 &l1,
       const point3 &l2, const point3 &s1,
       const point3 &s2, const point3 &s3){
2f24   point3 ret=pvec(s1,s2,s3);
38b0   double t=(ret.x*(s1.x-l1.x)+ret.y*(s1.y
         -l1.y)+ret.z*(s1.z-l1.z))/
e0f5     (ret.x*(l2.x-l1.x)+ret.y*(l2.y-l1.y)+
           ret.z*(l2.z-l1.z));
6302   ret.x=l1.x+(l2.x-l1.x)*t;
9cd9   ret.y=l1.y+(l2.y-l1.y)*t;
29b2   ret.z=l1.z+(l2.z-l1.z)*t;
ee0f   return ret;
95cf }
427e
427e // 计算两平面交线，注意事先判断是否平行，并
       保证三点不共线 ！
69e4 line3 intersection(const plane3 &u, const
        plane3 &v){
57af   line3 ret;
7b9a   ret.a=parallel(v.a,v.b,u.a,u.b,u.c)?
         intersection(v.b,v.c,u.a,u.b,u.c):
         intersection(v.a,v.b,u.a,u.b,u.c);
033e   ret.b=parallel(v.c,v.a,u.a,u.b,u.c)?
         intersection(v.b,v.c,u.a,u.b,u.c):
         intersection(v.c,v.a,u.a,u.b,u.c);
ee0f   return ret;
95cf }
68d7 line3 intersection(const point3 &u1,
       const point3 &u2, const point3 &u3,
       const point3 &v1, const point3 &v2,
       const point3 &v3){
57af   line3 ret;
0075   ret.a=parallel(v1,v2,u1,u2,u3)?
         intersection(v2,v3,u1,u2,u3):
         intersection(v1,v2,u1,u2,u3);
2a2b   ret.b=parallel(v3,v1,u1,u2,u3)?
         intersection(v2,v3,u1,u2,u3):
         intersection(v3,v1,u1,u2,u3);
ee0f   return ret;
95cf }
427e
427e // 点到直线距离
3696 double ptoline(const point3 &p, const
       line3 &l){
eddb   return len(xmult(p - l.a, l.b - l.a))/
         dis(l.a,l.b);
95cf }
c402 double ptoline(const point3 &p, const
       point3 &l1, const point3 &l2){
c833   return len(xmult(p - l1, l2 - l1))/dis(
         l1,l2);
95cf }
```

```
                                               427e
// 点到平面距离                                 427e
double ptoplane(const point3 &p, const         7ed7
  plane3 &s){
  return fabs(dmult(pvec(s), p - s.a))/        b87b
    len(pvec(s));
}                                              95cf
double ptoplane(const point3 &p, const         33d1
  point3 &s1, const point3 &s2, const
  point3 &s3){
  return fabs(dmult(pvec(s1,s2,s3), p -        3cea
    s1))/len(pvec(s1,s2,s3));
}                                              95cf
                                               427e
// 直线到直线距离                               427e
double linetoline(const line3 &u, const        eaed
  line3 &v){
  point3 n=xmult(u.a - u.b, v.a - v.b);        79e9
  return fabs(dmult(u.a - v.a,n))/len(n);      2a04
}                                              95cf
double linetoline(const point3 &u1, const      c288
   point3 &u2, const point3 &v1, const
  point3 &v2){
  point3 n=xmult(u1 - u2, v1 - v2);            a9a1
  return fabs(dmult(u1 - v1,n))/len(n);        04de
}                                              95cf
                                               427e
// 两直线夹角cos 值                             427e
double angle_cos(const line3 &u, const         7010
  line3 &v){
  return dmult(u.a - u.b, v.a - v.b)/len(      0567
    u.a - u.b)/len(v.a - v.b);
}                                              95cf
double angle_cos(const point3 &u1, const       a590
  point3 &u2, const point3 &v1, const
  point3 &v2){
  return dmult(u1 - u2, v1 - v2)/len(u1 -      b4cf
     u2)/len(v1 - v2);
}                                              95cf
                                               427e
// 两平面夹角cos 值                             427e
double angle_cos(const plane3 &u, const        5766
  plane3 &v){
  return dmult(pvec(u),pvec(v))/len(pvec(      8688
    u))/len(pvec(v));
}                                              95cf
double angle_cos(const point3 &u1, const       2053
  point3 &u2, const point3 &u3, const
  point3 &v1, const point3 &v2, const
  point3 &v3){
  return dmult(pvec(u1,u2,u3),pvec(v1,v2,      a583
    v3))/len(pvec(u1,u2,u3))/len(pvec(v1,
    v2,v3));
95cf }
427e
427e // 直线平面夹角sin 值
9344 double angle_sin(const line3 &l, const plane3 &s){
2177   return dmult(l.a - l.b,pvec(s))/len(l.a - l.b)/len(pvec(s));
95cf }
e36a double angle_sin(const point3 &l1,const point3 &l2, const point3 &s1, const point3 &s2, const point3 &s3){
14dc   return dmult(l1 - l2,pvec(s1,s2,s3))/len(l1 - l2)/len(pvec(s1,s2,s3));
95cf }
```

3.3 面积

```
427e // 求三角形有向表面积，输入三个顶点
0f35 double area_triangle(const plane3 &p){
c025   return len(xmult(p.b - p.a, p.c - p.a))/2;
95cf }
78b2 double area_triangle(const point3 &p1, const point3 &p2, const point3 &p3){
aa2a   return len(xmult(p2 - p1, p3 - p1))/2;
95cf }
6e8b double area_triangle(const point3 &p2, const point3 &p3){
c28a   return len(xmult(p2, p3))/2;
95cf }
427e // 求多边形有向表面积，输入三个顶点
2eca double area_polygon(int n,point3* p){
99ec   double s;
85c3   for (int i = 0;i < n; ++i)
d161     s += len(xmult(p[i], p[(i + 1) % n])) / 2;
fe09   return s;
95cf }
```

3.4 体积

```
427e // 求四面体有向体积，输入四个点
ef35 double volume_tetrahedron(const point3 &p1, const point3 &p2, const point3 &p3, const point3 &p4){
c2f2   return dmult(xmult(p1 - p4, p2 - p4), p3 - p4) / 6;
95cf }
b1cf double volume_tetrahedron(const point3 &p1, const point3 &p2, const point3 &p3){
74cc   return dmult(xmult(p1, p2), p3) / 6;
95cf }
double volume_tetrahedron(const plane3 &p){        2cde
  return dmult(xmult(p.a, p.b), p.c) / 6;          9493
}                                                  95cf
// 求多面体有向体积                                 427e
double volume_polygon(int n, plane3 *polygon){     0532
  double c = 0;                                    0ea9
  for (int i = 0; i < n; ++i)                      85c3
    c += volume_tetrahedron(polygon[i]);           8088
  return c;                                        14df
}                                                  95cf
```

3.5 重心

```
// 三角形重心                                                        427e
point3 barycenter(const point3 &a, const point3 &b, const point3 &c){  4ce9
  return a + b + c / 3;                                              85a3
}                                                                    95cf
// 四面体重心                                                        427e
point3 barycenter(const point3 &a, const point3 &b, const point3 &c, const point3 &d){  6485
  return (a + b) + (c + d) / 4;                                      3cb1
}                                                                    95cf
// 多面体重心                                                        427e
point3 barycenter(int n, plane3 *polygon){                           0a46
  point3 c;                                                          a1fc
  double v = 0;                                                      b0c2
  for (int i = 0; i < n; ++i){                                       6c2f
    double j = volume_tetrahedron(polygon[i]);                       e51b
    v += j;                                                          58e1
    c = c + (polygon[i].a + polygon[i].b + polygon[i].c) * j;        32b4
    }                                                                95cf
  return c / (4 * v);                                                432b
}                                                                    95cf
```

3.6 凸包

```
//a-b-右手定则指向凸包外面c        427e
const int MAXN = 500;              ce3a
const int MAXM = 250000;           9400
struct NODE{                       f8d9
  int p[4], next, out;             f5a3
  point3 f;                        7f4c
}s[MAXM];                          4cef
int edge[MAXN][MAXN];              91d4
int tot;                           8164
int next(int x){                   0019
  if (s[x].next == x) return x;    d010
901a    return s[x].next = next(s[x].next);
95cf  }
f01b  void add(int a, int b, int c, point3 *p){
582f    s[tot].p[0] = a;
661d    s[tot].p[1] = b;
50e3    s[tot].p[2] = c;
c4b1    s[tot].p[3] = a;
202c    s[tot].f = xmult(p[b] - p[a], p[c] - p[
          a]);
7fb3    s[tot].out = false;
100b    for (int i = 0; i < 3; ++i)
1e79      edge[s[tot].p[i]][s[tot].p[i + 1]] =
            tot;
ac2d    ++tot;
95cf  }
c1c6  void add(int a, int b, int c, int d,
        point3 *p){
05a3    point3 f = xmult(p[b] - p[a], p[c] - p[
          a]);
40d9    if (dmult(f, p[d] - p[a]) > 0) add(a, c
          , b, p);
3c28    else add(a, b, c, p);
95cf  }
427e
427e  // 主程序输入顶点个数，点集，返回面，
427e  // 要求不是所有点共面
4db6  int get_convex(int n, point3* p, plane3*
        convex){
d712    for (int i = 0; i < MAXM; ++i)
06de      s[i].next = i;
b7ad    tot = 0;
37f3    for (int i = 3; i < n; ++i)
79d4      if (!dots_onplane(p[0], p[1], p[2], p
            [i])){
3d14        swap(p[i], p[3]);
6173        break;
95cf      }
ea37    add(0, 1, 2, 3, p);
97fe    add(2, 3, 0, 1, p);
0393    add(3, 1, 0, 2, p);
cee5    add(3, 1, 2, 0, p);
7384    for (int i = 4; i < n; ++i){
e569      for (int j = next(0); j < tot; j =
            next(j + 1))
da36        s[j].out = dmult(s[j].f, p[i] - p[s
              [j].p[0]]) > 0;
21b5      int c = tot;
e569      for (int j = next(0); j < tot; j =
            next(j + 1))
9413        if (s[j].out){
9004          for (int k = 0; k < 3; ++k)
bda5            if (!s[edge[s[j].p[k + 1]][s[j
                  ].p[k]]].out)
            add(s[j].p[k], s[j].p[k + 1],      1807
               i, p);
        s[j].next = j + 1;                     68c3
      }                                        95cf
  }                                            95cf
  int i, j;                                    576f
  for (i = 0, j = next(0); j < tot; ++i,       d4d9
    j = next(j + 1)){
    convex[i].a = p[s[j].p[0]];                82ac
    convex[i].b = p[s[j].p[1]];                c1d7
    convex[i].c = p[s[j].p[2]];                f7bf
  }                                            95cf
  return i;                                    ffec
}                                              95cf
```

4 数论

4.1 头文件

```
// independent                                 427e
#include<iostream>                             e0a5
#include<algorithm>                            54ff
#include<cmath>                                c928
#include<cstdio>                               59b9
#include<cstdio>                               59b9
#include<map>                                  8c52
#include<set>                                  6326
#include<vector>                               09f7
#include<cstring>                              ef2f
#include<string>                               2349
using namespace std;                           421c
typedef long long LL;                          5cad
typedef unsigned int UI;                       1f2c
typedef unsigned long long ULL;                b773
const LL mod=1000000007;                       4d7e
```

4.2 基础

```
// independent                                 427e
// fast multiplication                         427e
LL FM(LL a,LL t,LL mod)                        3f6b
{                                              4506
  a%=mod;                                      af5c
  LL ans=1, mid=a;                             b56a
  while(t){                                    4c1b
    if(t&1) ans*=mid,ans%=mod;                 06fb
    mid*=mid;mid%=mod;                         3ea9
    t>>=1;                                     2f01
  }                                            95cf
  return ans;                                  4206
}                                              95cf
// gcd                                         427e
LL gcd(LL a,LL b)                              c2e9
```

```
4506 {
55d6   if (a<b){LL c=a;a=b;b=c;}
c56f   while (b!=0)
4506   {
28f6     LL c=a;a=b;b=c%b;
95cf   }
5ffd   return a;
95cf }
80eb LL lcm(LL a,LL b){
2dcf   LL g=gcd(a,b);
bda5   return a/g*b;
95cf }
427e // return gcd(a,b),a*x+b*y=g;
8534 LL ext_gcd(LL a,LL b,LL &x,LL &y)
4506 {
7d1a   if(b == 0){x = 1;y = 0;return a;}
e9fa   LL g = ext_gcd(b, a % b, x, y);
166e   LL t = x;x = y,y=t–a/b*y;
05da   return g;
95cf }
```

4.3 线性筛法

```
427e // independent
62a8 const int MAXN=10000000;
1858 int cprime[MAXN],used=0;
8a94 char p[MAXN];
0919 void prime_(){
ff7c    memset(p,0,sizeof p);
3099    for(int i=2;i<MAXN;i++){
f677      if(!p[i])  cprime[used]=i,used++;
97c2      for(int j=0;j<used;j++){
c98f        if(i*cprime[j]>MAXN) break;
ece6        p[i*cprime[j]]=true;
0f27        if(i%cprime[j]==0)break;
95cf      }
95cf    }
95cf }
```

4.4 线性同余方程

```
427e // depend on 基础：ext_gcd
427e
427e // 求一元线性同余方程： a*x = b mod m 的所
       有解（在同余系中）存于    ,数组ans 解的个数
       存在,中len
c8bd bool cong_eq(LL a,LL b,LL m,LL ans[],LL &
       len)
4506 {
2f15    LL g,x,y;
6b0f    g=ext_gcd(a,m,x,y);
1bd9    if(b%g) return false;
2800    LL base=((b/g*x)%m+m)%m;
62c8    len=g;
```

```
   for(int i=0;i<len;i++) ans[i]=(base+i     89a5
     *(m/len))%m;
   return true;                              3361
}                                            95cf
```

4.5 中国剩余定理

```
// depend on 基础：ext_gcd                    427e
                                             427e
// 中国剩余定理解特殊线性方程组，               427e
    x= a[i] mod b[i其中], b[i两两互质，
  共]个方程r
LL china(LL a[], LL b[], int r){             7300
  LL M=1;                                    afa3
  LL i,Mi,x0,y0,d,ans=0;                     8fa0
  for(i=0;i<r;i++){                          38d2
    M*=b[i];                                 b024
  }                                          95cf
  for(i=0;i<r;i++){                          38d2
    Mi=M/b[i];                               13d5
    ext_gcd(Mi,b[i],x0,y0);                  e90f
    ans=(ans+Mi*x0*a[i])%M;                  e55d
  }                                          95cf
  if(ans<0) ans+=M;                          361d
  return ans;                                4206
}                                            95cf
```

4.6 离散对数

```
// independent                               427e
// 求解形如  a^x= bmod MOD 的方程，，，已       427e
  知abMOD
                                             427e
const int maxn = 65535;                      ee1f
struct hash{                                 4609
  int a,b,next;                              db04
}Hash[maxn << 1];                            cbde
int flg[maxn];                               ab55
int top,idx;                                 3ce9
void ins(int a,int b){                       4921
  int k = b & maxn;                          2e3c
  if(flg[k] != idx){                         fcdc
    flg[k] = idx;                            29f2
    Hash[k].next = –1;                       81a6
    Hash[k].a = a;                           c3b7
    Hash[k].b = b;                           e19e
    return ;                                 4f2d
  }                                          95cf
  while(Hash[k].next != –1){                 fd3e
    if(Hash[k].b == b) return ;              e55c
    k = Hash[k].next;                        5551
  }                                          95cf
  Hash[k].next = ++ top;                     f7e2
  Hash[top].next = –1;                       8c95
```

```
d291   Hash[top].a = a;
ad12   Hash[top].b = b;
95cf }
99f4 int find(int b){
2e3c   int k = b & maxn;
a4dc   if(flg[k] != idx) return -1;
9030   while(k != -1){
b35f     if(Hash[k].b == b) return Hash[k].a;
5551     k = Hash[k].next;
95cf   }
fb5e   return -1;
95cf }
427e
e8bb int gcd(int a,int b){return b?gcd(b,a%b):
       a;}
5e78 int ext_gcd(int a,int b,int& x,int& y){
0c60   int t,ret;
a0bb   if (!b){x=1,y=0;return a;}
4d23   ret=ext_gcd(b,a%b,x,y);
0eb3   t=x,x=y,y=t-a/b*y;
ee0f   return ret;
95cf }
427e
49b5 int pow_mod(LL a,int b,int c)
4506 {
8fef   LL ret=1%c;a%=c;
7c06   while(b)
4506   {
6f75     if(b&1)
6bc0     ret=ret*a%c;
3386     a=a*a%c;
ca1f     b>>=1;
f959   }return ret;
95cf }
3be4 int Inval(int a,int b,int n){
9853   int x,y,e;
38ce   ext_gcd(a,n,x,y);
4c93   e=(LL)x*b%n;
4e9d   return e<0?e+n:e;
95cf }
c5f3 int BabyStep(int A,int B,int C){
856b   top = maxn; ++ idx;
c05a   LL buf=1%C,D=buf,K;
36c9   int i,d=0,tmp;
9f59   for(i=0;i<=100;buf=buf*A%C,++i)if(buf==
         B)return i;
87dc   while((tmp=gcd(A,C))!=1){
9506     if(B%tmp)return -1;
fb15     ++d;
6f10     C/=tmp;
e3f3     B/=tmp;
b8d3     D=D*A/tmp%C;
95cf   }
```

```
  int M=(int)ceil(sqrt((double)C));          fea4
  for(buf=1%C,i=0;i<=M;buf=buf*A%C,++i)      2662
    ins(i,buf);
  for(i=0,K=pow_mod((LL)A,M,C);i<=M;D=D*K    21a3
    %C,++i){
    tmp=Inval((int)D,B,C);int w ;            5d4e
    if(tmp>0&&(w = find(tmp)) != -1)         b3d8
      return i*M+w+d;
  }                                          95cf
  return -1;                                 fb5e
}                                            95cf
                                             427e
                                             427e
                                             427e
int main(){                                  3117
  int A,B,C;                                 1e17
  while(scanf("%d%d%d",&A,&C,&B)!=EOF,A      2072
    || B || C){
    B %= C;                                  1fc4
    int tmp=BabyStep(A,B,C);                 1719
    if(tmp<0)puts("No Solution");else        6a67
      printf("%d\n",tmp);
  }                                          95cf
  return 0;                                  7021
}                                            95cf
```

4.7 MillerRabin

```
// depend on 基础: multiMod                   427e
bool witness ( LL s , LL n ) {               44b6
  LL u = n - 1 ;                             c625
  int t = 0 ;                                2f70
  while ( ( u & 1 ) == 0 ) u >>= 1 , t ++    01e0
     ;
                                             427e
  LL x = FM ( s , u , n ) ;                  de44
  while ( t -- ) {                           3c2f
    LL tmp = x ;                             6da3
    x = multiMod ( x , x , n ) ;             7216
    if ( x == 1 ) {                          89ce
      if ( tmp == n - 1 || tmp == 1  )       a72c
        return false ;//may be prime
      else return true ;//composite          fe75
    }                                        95cf
  }                                          95cf
  return true ; //composite                  3361
}                                            95cf
                                             427e
bool millerRabin ( LL n , const int times    2daf
   = 3 ) {
  if ( n == 2 ) return true ;                89e8
  if ( ( n & 1 ) == 0 || n < 2 ) return      62ac
    false ;
  int i = times ;                            d8c4
148a    while ( i — ) {
2e40      LL s = rand ( ) % ( n – 1 ) + 1 ;
6474      if ( witness ( s , n ) ) return false
            ;
95cf    }
3361    return true ;
95cf  }
```

4.8 PollardRho

```
427e  // depend on 基础：，multiModFM
5bc3  LL multiMod ( LL a , LL b , LL n ) {
dcd8    a %= n ;
37f2    b %= n ;
9f3a    LL s = 0 ;
ca22    while( b ) {
90a9      if( b & 1 ) {
4134        s += a ;
f497        if( s >= n ) s –= n ;
95cf      }
80ab      a <<= 1 ; b >>= 1 ;
082b      if(a >= n ) a –= n ;
95cf    }
fe09    return  s ;
95cf  }
427e
9285  LL FM ( LL s , LL u , LL n ) {
c7db    s %= n ;
eb30    LL tmp = 1 ;
7ce6    while ( u ) {
5ab5      if ( u & 1 ) tmp = multiMod ( tmp , s
            , n ) ;
5085      s = multiMod ( s , s , n ) ;
517f      u >>= 1 ;
95cf    }
fe6e    return tmp ;
95cf  }
427e
44b6  bool witness ( LL s , LL n ) {
c625    LL u = n – 1 ;
2f70    int t = 0 ;
01e0    while ( ( u & 1 ) == 0 ) u >>= 1 , t ++
          ;
427e
de44    LL x = FM ( s , u , n ) ;
3c2f    while ( t — ) {
6da3      LL tmp = x ;
7216      x = multiMod ( x , x , n ) ;
89ce      if ( x == 1 ) {
a72c        if ( tmp == n – 1 || tmp == 1  )
              return false ;//may be prime
fe75        else return true ;//composite
95cf      }
95cf    }
```

```
  return true ; //composite                 3361
}                                           95cf
                                            427e
bool millerRabin ( LL n , const int times   2daf
   = 3 ) {
  if ( n == 2 ) return true ;               89e8
  if ( ( n & 1 ) == 0 || n < 2 ) return     62ac
    false ;
  int i = times ;                           d8c4
  while ( i — ) {                           148a
    LL s = rand ( ) % ( n – 1 ) + 1 ;       2e40
    if ( witness ( s , n ) ) return false   6474
      ;
  }                                         95cf
  return true ;                             3361
}                                           95cf
LL gcd ( LL a , LL b ) {                    4990
  if ( b == 0 ) return a ;                  22aa
  return gcd ( b , a % b ) ;                7b09
}                                           95cf
LL pollard_rho ( LL n ) {                   32db
  LL x , y , k , d  ;                       61ac
  x = y = rand ( ) % n ;                    2cce
  k = 2 ;                                   7924
  int i = 1 ;                               0d80
  int c =rand ( ) % n ;                     25c7
                                            427e
  while ( true ) {                          1026
    i ++ ;                                  a42b
    x = ( multiMod ( x , x , n ) + c ) %    a98e
      n ;
    if ( y == x ) return 1 ;//restart       abe5
    else if ( y > x ) d = gcd ( y – x , n   397a
      ) ;
    else d = gcd ( x – y , n ) ;            f73d
    if ( d != 1 && d != n – 1 ) return d    6dad
     ;
    else {                                  037f
      if ( i == k ) {                       679a
        y = x ;                             88e9
        k <<= 1 ;                           453c
      }                                     95cf
    }                                       95cf
  }                                         95cf
}存放分解出的质因子                          95cf
//                                          427e
LL factors [ 54 ] ;初始化为                  a6bf
//0                                         427e
int cnt ;                                   9f58
                                            427e
void split ( LL n ) {//n != 1               10b4
  if ( millerRabin ( n ) ) factors [ cnt    0494
    ++ ] = n ;
```

```
037f    else {
3942      LL p ;
a69f      do {
9a67        p = pollard_rho ( n ) ;
1ce0      }while ( p == n || p == 1 );
83ef      split ( p ) ;
0573      split ( n / p ) ;
95cf    }
95cf  }
```

4.9 矩阵基础

```
427e  // independent
cffb  struct Matrix{int m[MAXN][MAXN],l,r;
d184    Matrix(int w){
d0d9      l=r=w;
6b09      memset(m,0,sizeof m);
95cf    }
b54c    Matrix(){l=r=0;memset(m,0,sizeof m);}
329b  };
427e  // a.r = b.l
0a3f  Matrix operator * (Matrix a,Matrix b){
c97f    Matrix c;
3fe0    memset(c.m,0,sizeof c.m);
7272    c.l=a.l,c.r=b.r;
91ca    for(int i=0;i<a.l;i++){
6363      for(int j=0;j<b.r;j++){
d081        for(int k=0;k<a.r;k++){
78c3          c.m[i][j]+=a.m[i][k]*b.m[k][j];
427e  //        c.m[i][j]%=mod取余;//
95cf        }
95cf      }
95cf    }
14df    return c;
95cf  }
427e
427e  // a.l == b.l && a.r == b.r
ece1  Matrix operator + (Matrix a,Matrix b){
c97f    Matrix c;
b567    c.l=a.l;c.r=a.r;
91ca    for(int i=0;i<a.l;i++){
c58d      for(int j=0;j<a.r;j++){
c4cc        c.m[i][j]=a.m[i][j]+b.m[i][j];
bd48        c.m[i][j]%=mod;
95cf      }
95cf    }
14df    return c;
95cf  }
5eb9  Matrix FM(Matrix a,int t)
4506  {
d2a0    Matrix ans;ans.l=ans.r=a.l;
782d    for(int i=0;i<a.l;i++)ans.m[i][i]=1;
6880    Matrix mid=a;
4c1b    while(t){
```

```
      if(t&1) ans=ans*mid;                    e385
      mid=mid*mid;                            c1cb
      t>>=1;                                  2f01
    }                                         95cf
    return ans;                               4206
}                                             95cf
```

4.10 高斯消元

```
// independent                                427e
#define MAXN 100                              fb02
#define fabs(x) ((x)>0?(x):-(x))              c1b0
#define eps 1e-10列主元                        b76f
                                              427e
//消去求解gaussa[][]x[]=b[]返回是否有唯一解若有解在  427e
//,b中[]                                       427e
int gauss_cpivot(int n,double a[][MAXN],double b[]){  f2e9
  int i,j,k,row;                              c75e
  double maxp,t;                              ea2e
  for (k=0;k<n;k++){                          ab8e
    for (maxp=0,i=k;i<n;i++)                  a1ed
      if (fabs(a[i][k])>fabs(maxp))           8dd1
        maxp=a[row=i][k];                     f0ed
    if (fabs(maxp)<eps)                       ff5d
      return 0;                               7021
    if (row!=k){                              7dcf
      for (j=k;j<n;j++)                       0dff
        t=a[k][j],a[k][j]=a[row][j],a[row][j]=t;  b70f
      t=b[k],b[k]=b[row],b[row]=t;            243a
    }                                         95cf
    for (j=k+1;j<n;j++){                      1ff9
      a[k][j]/=maxp;                          8e4d
      for (i=k+1;i<n;i++)                     34eb
        a[i][j]-=a[i][k]*a[k][j];             56e2
    }                                         95cf
    b[k]/=maxp;                               4cd8
    for (i=k+1;i<n;i++)                       34eb
      b[i]-=b[k]*a[i][k];                     5fe8
  }                                           95cf
  for (i=n-1;i>=0;i--)                        e913
    for (j=i+1;j<n;j++)                       cd1d
      b[i]-=a[i][j]*b[j];                     2139
  return 1;                                   7459
}                                             95cf
```

5 数据结构

5.1 SplayTree

```
#include<cstdio>                              59b9
```

```
ef2f  #include<cstring>
bffa  #include<cstdlib>
427e
421c  using namespace std;
427e
1c79  #define MAXN 40010
06a1  #define INF 1ll<<62
5da6  #define MAX(a,b) ((a)>(b)?(a):(b))
11d0  struct SplayTree{
427e
f7cd    struct SplayNode{
b275      SplayNode *f,*C[2];
427e      //int s,ml,mr,max,tot,sz;
440f      long long s,c,sz;
427e      //bool rev,same;
b01b    } S[MAXN],*root,*null,*tr;
427e
4a30    int sz;
427e
88f1    void init()
4506    {
bd30      for (int i=0;i<=sz;i++) S[i].s=S[i].c
            =S[i].sz=0;
1bb9      sz=0;
47f6      null=NewNode(null,-INF);
ab2e      null->s=0;
f607      null->f=null;
c969      null->sz=0;
e3e8      null->C[0]=null->C[1]=null;
b6b3      root=NewNode(null,-INF);
0e03      root->C[1]=NewNode(root,INF);
2753      update(root->C[1]);
d657      update(root);
95cf    }
427e
b77a    SplayNode * NewNode( SplayNode *f, long
           long s){
dbed      SplayNode *ts;
40e5      ts=S+ ++sz;
2d59      ts->f=f;
bc22      ts->c=0;
577f      ts->C[0]=ts->C[1]=null;
ca6f      ts->s=s;
427e      //ts->tot=ts->max=ts->ml=ts->mr=ts->s
            ;
fdcf      ts->sz=1;
427e      //ts->rev=ts->same=0;
dd0c      return ts;
95cf    }
427e
9356    SplayTree(){
47f6      null=NewNode(null,-INF);
ab2e      null->s=0;
```

```
    null->f=null;                              f607
    null->sz=0;                                c969
    null->C[0]=null->C[1]=null;                e3e8
    root=NewNode(null,-INF);                   b6b3
    root->C[1]=NewNode(root,INF);              0e03
    update(root->C[1]);                        2753
    update(root);                              d657
  }                                            95cf
                                               427e
  void update( SplayNode * x){                 886d
    x->sz=x->C[0]->sz+x->C[1]->sz+1+x->c;      fa89
    /*x->tot=x->C[0]->tot+x->C[1]->tot+x       180a
      ->s;
    x->max=MAX(x->s,x->C[0]->max);             b048
    x->max=MAX(x->max,x->C[1]->max);           935a
    x->max=MAX(x->max,x->C[0]->mr+x->s);       3392
    x->max=MAX(x->max,x->C[1]->ml+x->s);       9a2e
    x->max=MAX(x->max,x->C[0]->mr+x->s+x       8176
      ->C[1]->ml);
    x->ml=MAX(x->C[0]->ml,x->C[0]->tot+x       0181
      ->s);
    x->ml=MAX(x->ml,x->C[0]->tot+x->s+x->      7a57
      C[1]->ml);
    x->mr=MAX(x->C[1]->mr,x->C[1]->tot+x       77d1
      ->s);
    x->mr=MAX(x->mr,x->C[1]->tot+x->s+x->      a71b
      C[0]->mr);*/
  }                                            95cf
                                               427e
  /*void labledown( SplayNode *x){             1455
    SplayNode *ts;                             dbed
    int tmp;                                   6eb3
    if (x==null||!(x->same||x->rev))           45c9
      return ;
    if (x->same){                              d66a
      x->C[1]->same=x->C[0]->same=1;           d26a
      x->C[1]->s=x->C[0]->s=x->s;              e573
      x->tot=x->s*x->sz;                       3058
      x->max=x->ml=x->mr=x->tot;               37f3
      if (x->s<0)                              89b1
        x->max=x->ml=x->mr=x->s;               be32
    }                                          95cf
    if (x->rev){                               e2e3
      tmp=x->ml;x->ml=x->mr;x->mr=tmp;         f24b
      ts=x->C[1];x->C[1]=x->C[0];x->C[0]=      ef88
        ts;
      x->C[1]->rev=!x->C[1]->rev;x->C          001e
        [0]->rev=!x->C[0]->rev;
    }                                          95cf
    x->same=x->rev=0;                          925b
  }*/                                          fe38
                                               427e
  void route( SplayNode *k1, int c){           965e
```

```
487c    SplayNode *k2=k1->f;
427e    //labledown(k2->C[!c]);labledown(k1->C[0]);labledown(k1->C[1]);
427e
18d3    k2->C[c]=k1->C[!c];
390f    k2->C[c]->f=k2;
8b3f    k1->f=k2->f;
70ea    if (k2->f->C[0]==k2) k2->f->C[0]=k1;
561f    else k2->f->C[1]=k1;
6439    k2->f=k1;
b627    k1->C[!c]=k2;
c05b    update(k2);//update(k1);
0e39    if (root==k2) root=k1;
95cf  }
427e
5237  SplayNode * rank( int k){
56b2    SplayNode *ts=root;
6eb3    int tmp;
1d6c    while (k){
427e      //labledown(ts);
188b      tmp=ts->C[0]->sz;
f57e      if (k<=tmp) ts=ts->C[0];
17be      else if (k<=tmp+ts->c+1) break;
f19e      else k-=tmp+ts->c+1,ts=ts->C[1];
95cf    }
dd0c    return ts;
95cf  }
427e
ce71  /*void select( int s, int r){
46d4    rank(s,null);rank(r,root);
fe38  }*/
427e
58b9  void splay( SplayNode *x, SplayNode *s){
961d    if (x==null) return ;
66e8    update(x);
427e    //labledown(x);
ea33    while (x->f!=s){
b6c4      if (x->f->f==s){
2f96        if (x->f->C[0]==x)
5c86          route(x,0);
649a        else
f931          route(x,1);
1112      } else if (x->f->f->C[0]==x->f){
2f96        if (x->f->C[0]==x)
3252          route(x->f,0),route(x,0);
649a        else
b8c5          route(x,1),route(x,0);
8e2e      } else {
717f        if (x->f->C[1]==x)
eb10          route(x->f,1),route(x,1);
649a        else
142a          route(x,0),route(x,1);
```

```
    }                                    95cf
  }                                      95cf
      update(x);                         66e8
}                                        95cf
                                         427e
void ins( long long k)                   519b
{                                        4506
  SplayNode *ts=root,*ls=null;           3cd1
  while (ts!=null){                      8d36
    //labledown(ts);                     427e
    ls=ts;                               17d7
    if (k==ts->s)                        4be3
    {                                    4506
      ts->c++;                           ebae
      update(ts);                        841d
      splay(ts,null);                    530f
      return ;                           4f2d
    }                                    95cf
    else if (k<ts->s) ts=ts->C[0];       35a1
    else ts=ts->C[1];                    b87e
  }                                      95cf
  if (k<ls->s)                           4efc
  {                                      4506
    ls->C[0]=NewNode(ls,k);              7728
    update(ls->C[0]);update(ls);         96ed
    splay(ls->C[0],null);                4b8a
  } else                                 d268
  {                                      4506
    ls->C[1]=NewNode(ls,k);              703f
    update(ls->C[1]);update(ls);         9628
    splay(ls->C[1],null);                758e
  }                                      95cf
}                                        95cf
                                         427e
//value k                                427e
void remove( long long k)                6bd3
{                                        4506
  SplayNode *ts=root,*ls=null;           3cd1
  while (ts!=null){                      8d36
    //labledown(ts);                     427e
    ls=ts;                               17d7
    if (k==ts->s)                        4be3
    {                                    4506
      if (ts->c==0)                      c447
      {                                  4506
        splay(ts,null);                  530f
        splay(rank(ts->C[0]->sz),root);  77f9
        root=ts->C[0];                   646e
        root->f=null;                    d16b
        root->C[1]=ts->C[1];             469b
        ts->C[1]->f=root;                1105
        update(root);                    d657
      } else ts->c--,update(ts),splay(   3f99
                ts,null);
4f2d          return ;
95cf        }
35a1        else if (k<ts->s) ts=ts->C[0];
b87e        else ts=ts->C[1];
95cf      }
4f2d      return ;//not find
95cf    }
427e
427e    //rank k
0909    void del( int k)
4506    {
d827      splay(rank(k+1),null);
79ab      splay(rank(root->C[0]->sz),root);
56b2      SplayNode *ts=root;
646e      root=ts->C[0];
d16b      root->f=null;
469b      root->C[1]=ts->C[1];
1105      ts->C[1]->f=root;
d657      update(root);
95cf    }
427e
d16f    int find( int k)
4506    {
d827      splay(rank(k+1),null);
1ff2      return root->s;
95cf    }
427e
8c3f    int find_v( int k)
4506    {
56b2      SplayNode *ts=root;
8d36      while (ts!=null){
427e        //labledown(ts);
4be3        if (k==ts->s)
4506        {
530f          splay(ts,null);
7459          return 1;
95cf        }
35a1        else if (k<ts->s) ts=ts->C[0];
b87e        else ts=ts->C[1];
95cf      }
7021      return 0;
95cf    }
427e
4a3d    bool empty()
4506    {
c635      return !(root->sz-2>0);
95cf    }
eb81  } T;
427e
427e
59b9  #include<cstdio>
ef2f  #include<cstring>
```

```
#include<cstdlib>                                    bffa
#include<algorithm>                                  54ff
#include<queue>                                      acb9
                                                     427e
using namespace std;                                 421c
                                                     427e
priority_queue<int> MinNumber;                       baf7
                                                     427e
#define MAXN 300010                                  b47b
#define INF 1ll<<62                                  06a1
#define MAX(a,b) ((a)>(b)?(a):(b))                   5da6
                                                     427e
                                                     427e
                                                     427e
struct SplayNode{                                    f7cd
  SplayNode *f,*C[2];                                b275
  //int s,ml,mr,max,tot,sz;                          427e
  long long s,c,sz,tot;                              880b
  int neg,pos,sta;                                   9673
  //bool rev,same;                                   427e
};                                                   329b
                                                     427e
int N,st,it;                                         e929
char s[100];                                         6c85
SplayNode *A[MAXN],*B[MAXN];                         437d
                                                     427e
                                                     427e
struct SplayTree{                                    11d0
                                                     427e
  int sz;                                            4a30
  SplayNode S[MAXN],*root,*null,*tr;                 ea48
                                                     427e
  void init()                                        88f1
  {                                                  4506
    for (int i=0;i<=sz;i++) S[i].s=S[i].c            bd30
      =S[i].sz=0;
    sz=0;                                            1bb9
    null=NewNode(null,-INF);                         47f6
    null->s=0;                                       ab2e
    null->tot=0;                                     6b1f
    null->neg=null->pos=0;                           0380
    null->f=null;                                    f607
    null->sz=0;                                      c969
    null->C[0]=null->C[1]=null;                      e3e8
    root=NewNode(null,0);                            8259
    root->C[1]=NewNode(root,0);                      6a6a
    root->neg=root->pos=root->C[1]->neg=             a562
      root->C[0]->pos=root->sta=root->C
      [1]->sta=0;
    update(root->C[1]);                              2753
    update(root);                                    d657
  }                                                  95cf
                                                     427e
```

```
b77a  SplayNode * NewNode( SplayNode *f, long long s){
dbed    SplayNode *ts;
40e5    ts=S+ ++sz;
2d59    ts->f=f;
577f    ts->C[0]=ts->C[1]=null;
ca6f    ts->s=s;
2d6b    ts->tot=s;
bd5a    ts->neg=s<0?1:0;
ac50    ts->pos=s>0?1:0;
b142    ts->sta=s>0?1:-1;
fdcf    ts->sz=1;
dd0c    return ts;
95cf  }
427e
9356  SplayTree(){
47f6    null=NewNode(null,-INF);
ab2e    null->s=0;
f607    null->f=null;
c969    null->sz=0;
e3e8    null->C[0]=null->C[1]=null;
8259    root=NewNode(null,0);
6a6a    root->C[1]=NewNode(root,0);
2753    update(root->C[1]);
d657    update(root);
95cf  }
427e
886d  void update( SplayNode * x){
fa89    x->sz=x->C[0]->sz+x->C[1]->sz+1+x->c;
7619    x->tot=x->C[0]->tot+x->C[1]->tot+x->s;
9fe9    x->neg=x->C[0]->neg+x->C[1]->neg+(x->sta==-1?1:0);
fb82    x->pos=x->C[0]->pos+x->C[1]->pos+(x->sta==1?1:0);
95cf  }
427e
965e  void route( SplayNode *k1, int c){
487c    SplayNode *k2=k1->f;
427e    //labledown(k2->C[!c]);labledown(k1->C[0]);labledown(k1->C[1]);
427e
18d3    k2->C[c]=k1->C[!c];
390f    k2->C[c]->f=k2;
8b3f    k1->f=k2->f;
70ea    if (k2->f->C[0]==k2) k2->f->C[0]=k1;
561f    else k2->f->C[1]=k1;
6439    k2->f=k1;
b627    k1->C[!c]=k2;
f5eb    update(k2);update(k1);
0e39    if (root==k2) root=k1;
95cf  }
427e
```

```
  SplayNode * rank( int k){                             5237
    SplayNode *ts=root;                                 56b2
    int tmp;                                            6eb3
    while (k){                                          1d6c
//        labledown(ts);                                427e
      tmp=ts->C[0]->sz;                                 188b
      if (k<=tmp) ts=ts->C[0];                          f57e
      else if (k<=tmp+1) break;                         f1b2
      else k-=tmp+1,ts=ts->C[1];                        fefe
    }                                                   95cf
    return ts;                                          dd0c
  }                                                     95cf
                                                        427e
                                                        427e
  long long query( SplayNode *l, SplayNode *r)          f257
  {                                                     4506
    splay(l,null);splay(r,root);                        da8a
    return root->C[1]->C[0]->tot+root->s+root->C[1]->s; 7602
  }                                                     95cf
                                                        427e
  void splay( SplayNode *x, SplayNode *s) {             58b9
    if (x==null) return ;                               961d
    update(x);                                          66e8
    //labledown(x);                                     427e
    while (x->f!=s){                                    ea33
      if (x->f->f==s){                                  b6c4
        if (x->f->C[0]==x)                              2f96
          route(x,0);                                   5c86
        else                                            649a
          route(x,1);                                   f931
      } else if (x->f->f->C[0]==x->f){                  1112
        if (x->f->C[0]==x)                              2f96
          route(x->f,0),route(x,0);                     3252
        else                                            649a
          route(x,1),route(x,0);                        b8c5
      } else {                                          8e2e
        if (x->f->C[1]==x)                              717f
          route(x->f,1),route(x,1);                     eb10
        else                                            649a
          route(x,0),route(x,1);                        142a
      }                                                 95cf
    }                                                   95cf
        update(x);                                      66e8
  }                                                     95cf
                                                        427e
  //rank k                                              427e
  void del( SplayNode *ss)                              aae1
  {                                                     4506
    splay(ss,null);                                     fc5e
    splay(rank(root->C[0]->sz),root);                   79ab
56b2     SplayNode *ts=root;
646e     root=ts->C[0];
d16b     root->f=null;
469b     root->C[1]=ts->C[1];
1105     ts->C[1]->f=root;
d657     update(root);
95cf   }
427e
0edf   void ins( int k, int s)
4506   {
d827     splay(rank(k+1),null);
427e     //splay(rank(k),root);
e7e2     SplayNode *ts=NewNode(root,s);
7235     root->C[0]->f=ts;
3fe8     ts->C[0]=root->C[0];
d5f0     root->C[0]=ts;
f9af     A[s]=ts;
530f     splay(ts,null);
95cf   }
a228   void find( int t)
4506   {
af72     SplayNode *ts=root,*ls;
eb9a     while (ts!=null)
4506     {
17d7       ls=ts;
eec3       if (ts->C[1]->neg>=t) ts=ts->C[1];
20e3       else if (ts->C[1]->neg+(ts->sta
             <0?1:0)==t) break;
3c94       else t-=ts->C[1]->neg+(ts->sta
             <0?1:0),ts=ts->C[0];
95cf     }
8740     if (ts==null)splay(ls,null);
9bfe     else splay(ts,null);
95cf   }
df17   void insins( int k, int s)
4506   {
9131     int t=root->C[1]->pos;
b7b0     find(t);
ee3d     SplayNode *ts=NewNode(root,-s);
7235     root->C[0]->f=ts;
3fe8     ts->C[0]=root->C[0];
d5f0     root->C[0]=ts;
ebf4     B[s]=ts;
530f     splay(ts,null);
95cf   }
eb81 } T;
427e
0edf void ins( int k, int s)
4506 {
b69b   T.ins(k,s);
0a72   T.insins(k,s);
95cf }
427e
```

```
                                             427e
int main()                                   299c
{                                            4506
  int cases=0;                               1008
  while (scanf("%d\n",&N)!=EOF)              b889
  {                                          4506
    T.init();                                db73
    while (MinNumber.size()) MinNumber.      19f9
      pop();
    int nown=0,t;                            6b7c
    printf("Case #%d:\n",++cases);           524a
    while (N--)                              720b
    {                                        4506
      scanf("%s %d\n",s,&t);                 860a
      if (s[0]=='i')                         b81b
      {                                      4506
        t++;                                 5006
        if (MinNumber.empty()) it=++nown;    7309
        else it=-MinNumber.top(),            c2a0
          MinNumber.pop();
        ins(t,it);                           a3af
      } else if (s[0]=='r')                  603e
      {                                      4506
        MinNumber.push(-t);                  3834
        T.del(A[t]);                         bf62
        T.del(B[t]);                         bb40
      } else if (s[0]=='q')                  b72a
      {                                      4506
        printf("%I64d\n",T.query(A[t],B[t    0016
          ]));
      }                                      95cf
    }                                        95cf
  }                                          95cf
}                                            95cf
```

5.2 SplayTree

```
#include<cstdio>                             59b9
#include<cstring>                            ef2f
#include<cstdlib>                            bffa
                                             427e
using namespace std;                         421c
                                             427e
#define MAXN 40010                           1c79
#define INF 1ll<<62                          06a1
#define MAX(a,b) ((a)>(b)?(a):(b))           5da6
struct SplayTree{                            11d0
                                             427e
  struct SplayNode{                          f7cd
    SplayNode *f,*C[2];                      b275
    //int s,ml,mr,max,tot,sz;                427e
    long long s,c,sz;                        440f
    //bool rev,same;                         427e
  } S[MAXN],*root,*null,*tr;                 b01b
```

```
427e
4a30 int sz;
427e
88f1 void init()
4506 {
bd30   for (int i=0;i<=sz;i++) S[i].s=S[i].c=S[i].sz=0;
1bb9   sz=0;
47f6   null=NewNode(null,-INF);
ab2e   null->s=0;
f607   null->f=null;
c969   null->sz=0;
e3e8   null->C[0]=null->C[1]=null;
b6b3   root=NewNode(null,-INF);
0e03   root->C[1]=NewNode(root,INF);
2753   update(root->C[1]);
d657   update(root);
95cf }
427e
b77a SplayNode * NewNode( SplayNode *f, long long s){
dbed   SplayNode *ts;
40e5   ts=S+ ++sz;
2d59   ts->f=f;
bc22   ts->c=0;
577f   ts->C[0]=ts->C[1]=null;
ca6f   ts->s=s;
427e   //ts->tot=ts->max=ts->ml=ts->mr=ts->s;
fdcf   ts->sz=1;
427e   //ts->rev=ts->same=0;
dd0c   return ts;
95cf }
427e
9356 SplayTree(){
47f6   null=NewNode(null,-INF);
ab2e   null->s=0;
f607   null->f=null;
c969   null->sz=0;
e3e8   null->C[0]=null->C[1]=null;
b6b3   root=NewNode(null,-INF);
0e03   root->C[1]=NewNode(root,INF);
2753   update(root->C[1]);
d657   update(root);
95cf }
427e
886d void update( SplayNode * x){
fa89   x->sz=x->C[0]->sz+x->C[1]->sz+1+x->c;
180a   /*x->tot=x->C[0]->tot+x->C[1]->tot+x->s;
b048   x->max=MAX(x->s,x->C[0]->max);
935a   x->max=MAX(x->max,x->C[1]->max);
3392   x->max=MAX(x->max,x->C[0]->mr+x->s);
```

```
  x->max=MAX(x->max,x->C[1]->ml+x->s);                9a2e
  x->max=MAX(x->max,x->C[0]->mr+x->s+x->C[1]->ml);    8176
  x->ml=MAX(x->C[0]->ml,x->C[0]->tot+x->s);           0181
  x->ml=MAX(x->ml,x->C[0]->tot+x->s+x->C[1]->ml);     7a57
  x->mr=MAX(x->C[1]->mr,x->C[1]->tot+x->s);           77d1
  x->mr=MAX(x->mr,x->C[1]->tot+x->s+x->C[0]->mr);*/   a71b
}                                                     95cf
                                                      427e
/*void labledown( SplayNode *x){                      1455
  SplayNode *ts;                                      dbed
  int tmp;                                            6eb3
  if (x==null||!(x->same||x->rev)) return ;           45c9
  if (x->same){                                       d66a
    x->C[1]->same=x->C[0]->same=1;                    d26a
    x->C[1]->s=x->C[0]->s=x->s;                       e573
    x->tot=x->s*x->sz;                                3058
    x->max=x->ml=x->mr=x->tot;                        37f3
    if (x->s<0)                                       89b1
      x->max=x->ml=x->mr=x->s;                        be32
  }                                                   95cf
  if (x->rev){                                        e2e3
    tmp=x->ml;x->ml=x->mr;x->mr=tmp;                  f24b
    ts=x->C[1];x->C[1]=x->C[0];x->C[0]=ts;            ef88
    x->C[1]->rev=!x->C[1]->rev;x->C[0]->rev=!x->C[0]->rev;  001e
  }                                                   95cf
  x->same=x->rev=0;                                   925b
}*/                                                   fe38
                                                      427e
void route( SplayNode *k1, int c){                    965e
  SplayNode *k2=k1->f;                                487c
  //labledown(k2->C[!c]);labledown(k1->C[0]);labledown(k1->C[1]);  427e
                                                      427e
  k2->C[c]=k1->C[!c];                                 18d3
  k2->C[c]->f=k2;                                     390f
  k1->f=k2->f;                                        8b3f
  if (k2->f->C[0]==k2) k2->f->C[0]=k1;                70ea
  else k2->f->C[1]=k1;                                561f
  k2->f=k1;                                           6439
  k1->C[!c]=k2;                                       b627
  update(k2);//update(k1);                            c05b
  if (root==k2) root=k1;                              0e39
}                                                     95cf
                                                      427e
SplayNode * rank( int k){                             5237
```

```
56b2    SplayNode *ts=root;
6eb3    int tmp;
1d6c    while (k){
427e      //labledown(ts);
188b      tmp=ts->C[0]->sz;
f57e      if (k<=tmp) ts=ts->C[0];
17be      else if (k<=tmp+ts->c+1) break;
f19e      else k-=tmp+ts->c+1,ts=ts->C[1];
95cf    }
dd0c    return ts;
95cf  }
427e
ce71  /*void select( int s, int r){
46d4    rank(s,null);rank(r,root);
fe38  }*/
427e
58b9  void splay( SplayNode *x, SplayNode *s)
        {
961d    if (x==null) return ;
66e8    update(x);
427e    //labledown(x);
ea33    while (x->f!=s){
b6c4      if (x->f->f==s){
2f96        if (x->f->C[0]==x)
5c86          route(x,0);
649a        else
f931          route(x,1);
1112      } else if (x->f->f->C[0]==x->f){
2f96        if (x->f->C[0]==x)
3252          route(x->f,0),route(x,0);
649a        else
b8c5          route(x,1),route(x,0);
8e2e      } else {
717f        if (x->f->C[1]==x)
eb10          route(x->f,1),route(x,1);
649a        else
142a          route(x,0),route(x,1);
95cf      }
95cf    }
66e8        update(x);
95cf  }
427e
519b  void ins( long long k)
4506  {
3cd1    SplayNode *ts=root,*ls=null;
8d36    while (ts!=null){
427e      //labledown(ts);
17d7      ls=ts;
4be3      if (k==ts->s)
4506      {
ebae        ts->c++;
841d        update(ts);
530f        splay(ts,null);
```

```
      return ;                             4f2d
    }                                      95cf
    else if (k<ts->s) ts=ts->C[0];         35a1
    else ts=ts->C[1];                      b87e
  }                                        95cf
  if (k<ls->s)                             4efc
  {                                        4506
    ls->C[0]=NewNode(ls,k);                7728
    update(ls->C[0]);update(ls);           96ed
    splay(ls->C[0],null);                  4b8a
  } else                                   d268
  {                                        4506
    ls->C[1]=NewNode(ls,k);                703f
    update(ls->C[1]);update(ls);           9628
    splay(ls->C[1],null);                  758e
  }                                        95cf
}                                          95cf
                                           427e
//value k                                  427e
void remove( long long k)                  6bd3
{                                          4506
  SplayNode *ts=root,*ls=null;             3cd1
  while (ts!=null){                        8d36
    //labledown(ts);                       427e
    ls=ts;                                 17d7
    if (k==ts->s)                          4be3
    {                                      4506
      if (ts->c==0)                        c447
      {                                    4506
        splay(ts,null);                    530f
        splay(rank(ts->C[0]->sz),root);    77f9
        root=ts->C[0];                     646e
        root->f=null;                      d16b
        root->C[1]=ts->C[1];               469b
        ts->C[1]->f=root;                  1105
        update(root);                      d657
      } else ts->c--,update(ts),splay(     3f99
        ts,null);
      return ;                             4f2d
    }                                      95cf
    else if (k<ts->s) ts=ts->C[0];         35a1
    else ts=ts->C[1];                      b87e
  }                                        95cf
  return ;//not find                       4f2d
}                                          95cf
                                           427e
//rank k                                   427e
void del( int k)                           0909
{                                          4506
  splay(rank(k+1),null);                   d827
  splay(rank(root->C[0]->sz),root);        79ab
  SplayNode *ts=root;                      56b2
  root=ts->C[0];                           646e
```

```
d16b     root->f=null;
469b     root->C[1]=ts->C[1];
1105     ts->C[1]->f=root;
d657     update(root);
95cf   }
427e
d16f   int find( int k)
4506   {
d827     splay(rank(k+1),null);
1ff2     return root->s;
95cf   }
427e
8c3f   int find_v( int k)
4506   {
56b2     SplayNode *ts=root;
8d36     while (ts!=null){
427e       //labledown(ts);
4be3       if (k==ts->s)
4506       {
530f         splay(ts,null);
7459         return 1;
95cf       }
35a1       else if (k<ts->s) ts=ts->C[0];
b87e       else ts=ts->C[1];
95cf     }
7021     return 0;
95cf   }
427e
4a3d   bool empty()
4506   {
c635     return !(root->sz-2>0);
95cf   }
eb81 } T;
```

5.3 kdtree

```
0193 const int inf = 1000000000;
f877 #define sqr(x) (((long long)(x))*(x))
7829 const int MAXN = 500000;
427e //MAXM 维度数
d975 const int MAXM = 2;
427e // 必须手动设置点的度数 degree
5f09 struct POINT{
a5e8   int x[MAXM], lx[MAXM], rx[MAXM];
082e   int index;
7864   int degree;
aa08   POINT(){
3b30     for (int i = 0; i < MAXM; ++i){
8339       lx[i] = 0;
58a4       rx[i] = inf;
95cf     }
329b   };
9bba   long long dis(const POINT &a){
889f     long long ans = 0;
b02e     for (int i = 0; i < degree; ++i)
d902       ans += sqr(x[i] - a.x[i]);
4206     return ans;
95cf   }
c188 }p[MAXN];
427e //comindex 是首先比较的维度必须在比较之前赋值
6343 int cmpindex;
f78e bool cmp(const POINT &a, const POINT &b){
ceb8   for (int i = 0; i < a.degree; ++i){
6055     int j = (i + cmpindex) % a.degree;
2abf     if (a.x[j] != b.x[j])
d66a       return a.x[j] < b.x[j];
95cf   }
438e   return false;
95cf }
427e //degree 首先被比较的维度
f8d9 struct NODE{
38a5   POINT p;
6f63   int left, right, father;
a55b   int number, degree, flag;
c024   void init(const POINT &a, int d){
0c52     p = a;
009e     left = -1;
5e06     right = -1;
fae0     degree = d;
b556     number = 1;
3a4a     flag = false;
95cf   }
96c3 }node[MAXN];
6ee4 int nodesize;
427e // 储存结果
eea1 struct CYL{
38a5   POINT p;
eae0   long long r;
2db0   bool operator < (const CYL &a)const{
c9e2     return r < a.r;
95cf   }
329b };
427e // 求出v 节点下的所有点到顶点p 的距离
4e6c void searchr(int v, const POINT &p, priority_queue<CYL> &pq, int k){
d043   if (v == -1) return;
0109   CYL c;
add6   c.p = node[v].p;
ffe7   c.r = c.p.dis(p);
b544   pq.push(c);
e49b   while (pq.size() > k) pq.pop();
5ceb   searchr(node[v].left, p, pq, k);
6d98   searchr(node[v].right, p, pq, k);
95cf }
427e // 贪心地返回点数接近k 的节点
48d3 int searchknode(int v, const POINT &p,
       int k){
1cc2   if ((v == -1)||(node[v].number < k))
         return -1;
d02f   cmpindex = node[v].degree;
ff78   if (cmp(p, node[v].p)){
e7b8     if ((node[v].left != -1)&&(node[node[
           v].left].number >= k))
006e       return searchknode(node[v].left, p,
             k);
8e2e   }else{
5fcc     if ((node[v].right != -1)&&(node[node
           [v].right].number >= k))
527c       return searchknode(node[v].right, p
             , k);
95cf   }
aa78   return v;
95cf }
427e // 检查某区域是否可能有k 小的点
3ef1 bool check(POINT &root, const POINT &p,
       priority_queue<CYL> &pq){
d3df   POINT c;
ad9e   c.degree = p.degree;
af4d   for (int i = 0; i < p.degree; ++i){
2451     if ((root.lx[i] <= p.x[i])&&(p.x[i]
           <= root.rx[i])){
91b7       c.x[i] = p.x[i];
b333       continue;
95cf     }
b3f8     if (p.x[i] < root.lx[i]) c.x[i] =
           root.lx[i];
0817     else c.x[i] = root.rx[i];
95cf   }
0f33   if (c.dis(p) < pq.top().r) return true;
438e   return false;
95cf }
427e // 寻找距离最近的k 个点
0fb3 void findk(int v, const POINT &p, int k,
       priority_queue<CYL> &pq){
5dd1   if (node[v].flag){
e213     node[v].flag = false;
4f2d     return;
95cf   }
af9c   long long d = node[v].p.dis(p);
0b50   if (d < pq.top().r){
0109     CYL c;
add6     c.p = node[v].p;
895b     c.r = d;
b544     pq.push(c);
4e19     pq.pop();
95cf   }
f7cb   if ((node[v].left != -1)&&check(node[
         node[v].left].p, p, pq))
2836     findk(node[v].left, p, k, pq);
```

```
f1f4   if ((node[v].right != -1)&&check(node[
         node[v].right].p, p, pq))
5846     findk(node[v].right, p, k, pq);
95cf }
427e //KNN 算法，输入KD-tree 的根，返回与 p 最相
       邻的k 个点
427e //ans 中保证点到的距离依次递增p
4d4d void KNN(const POINT &p, int k, POINT ans
       []){
b335   priority_queue<CYL> pq;
3c91   int v = searchknode(0, p, k);
427e   //cout<<v<<endl;
13c5   node[v].flag = true;
c275   searchr(v, p, pq, k);
aa47   while(pq.size() > k){
4e19     pq.pop();
95cf   }
8766   cnt = 0;
07c2   findk(0, p, k, pq);
e213   node[v].flag = false;
9bd2   int n = pq.size();
130b   for (int i = 0; i < k; ++i){
bcd6     ans[i] = pq.top().p;
4e19     pq.pop();
95cf   }
2741   reverse(ans, ans + k);
95cf }
427e // 设置每个区域的界
007d void boundset(int v){
d043   if (v == -1) return;
1c34   if (node[v].left != -1){
788a     for (int i = 0; i < node[v].p.degree;
            ++i){
4cd0       node[node[v].left].p.rx[i] = node[v
             ].p.rx[i];
01cc       node[node[v].left].p.lx[i] = node[v
             ].p.lx[i];
95cf     }
1cf1     node[node[v].left].father = v;
2de3     node[node[v].left].p.rx[node[v].
           degree] = node[v].p.x[node[v].
           degree];
9123     boundset(node[v].left);
95cf   }
3d5d   if (node[v].right != -1){
788a     for (int i = 0; i < node[v].p.degree;
            ++i){
9cb6       node[node[v].right].p.rx[i] = node[
             v].p.rx[i];
1388       node[node[v].right].p.lx[i] = node[
             v].p.lx[i];
95cf     }
51dc     node[node[v].right].father = v;
910e      node[node[v].right].p.lx[node[v].
            degree] = node[v].p.x[node[v].
            degree];
33ad      boundset(node[v].right);
95cf    }
95cf  }
427e
427e  // 根据p 中的[s, t) 点建立Kd–TREE ,
427e  //p 中的顺序会被破坏
427e  //degree 是首先比较的维度
d02f  int buildtree(int s, int t, int degree,
        POINT p[], begin = 1){
fb76    if (begin) nodesize = 0;
636f    if (s == t) return –1;
34f1    else if(t – s == 1){
e0a6      node[nodesize].init(p[s], degree);
11a5      return nodesize++;
95cf    }
910c    cmpindex = degree;
4aed    sort(p + s, p + t, cmp);
a7b0    int mid = (s + t – 1) >> 1;
7b7e    int v = nodesize;
8ce7    node[nodesize++].init(p[mid], degree);
ac44    node[v].left = buildtree(s, mid, (
          degree + 1) % p[mid].degree, p, 0);
b4f2    node[v].right = buildtree(mid + 1, t, (
          degree + 1) % p[mid].degree, p, 0);
6a2a    node[v].number = t – s;
e8ae    if (!v){
74a5      node[0].father = –1;
0176      boundset(0);
95cf    }
aa78    return v;
95cf  }
```

5.4 后缀数组

```
e0a5  #include<iostream>
59b9  #include<cstdio>
09f7  #include<vector>
ef2f  #include<cstring>
2349  #include<string>
421c  using namespace std;
f7d6  const int maxn=10000;
1779  #define F(x) ((x)/3+((x)%3==1?0:tb))
2358  #define G(x) ((x)<tb?(x)*3+1:((x)–tb)
        *3+2)值得注意的
        是，
427e
427e  //数组rank 与数组互逆 sa ，所以可以根据sa 在
        （）时间内求出 Onrank以下为倍增算法实现
427e  // （） Onlogn
427e  //r 为字符串数组  ,sa 为结果S 的n 个后缀从
        小到大进行排序之后把排好序的后缀的开头位置
        顺序，也就是字典序次放入
//sa 中                                        427e
                                               427e
int wa[maxn],wb[maxn],wv[maxn],ws_[maxn];      2278
int cmp(int *r,int a,int b,int l)              3aa5
{                                              4506
  return r[a]==r[b]&&r[a+l]==r[b+l];           e543
}                                              95cf
//是数组的长度，即（）nrstrlenr,m 为语言集的      427e
  最大编号
void da(int *r,int *sa,int n,int m)            cdcb
{                                              4506
   int i,j,p,*x=wa,*y=wb,*t;                   8fa8
   for(i=0;i<m;i++) ws_[i]=0;                  c934
   for(i=0;i<n;i++) ws_[x[i]=r[i]]++;          e6d3
   for(i=1;i<m;i++) ws_[i]+=ws_[i–1];          2d9a
   for(i=n–1;i>=0;i––) sa[––ws_[x[i]]]=i;      5b0a
                                               427e
   for(j=1,p=1;p<n;j*=2,m=p)                   efbc
   {                                           4506
    for(p=0,i=n–j;i<n;i++) y[p++]=i;           5b7b
    for(i=0;i<n;i++) if(sa[i]>=j) y[p++]=      dc1c
      sa[i]–j;
    for(i=0;i<n;i++) wv[i]=x[y[i]];            792f
    for(i=0;i<m;i++) ws_[i]=0;                 c934
    for(i=0;i<n;i++) ws_[wv[i]]++;             f297
    for(i=1;i<m;i++) ws_[i]+=ws_[i–1];         2d9a
    for(i=n–1;i>=0;i––) sa[––ws_[wv[i]]]=      e91d
      y[i];
    for(t=x,x=y,y=t,p=1,x[sa[0]]=0,i=1;i<      0b94
      n;i++)
     x[sa[i]]=cmp(y,sa[i–1],sa[i],j)?p–1:      fef3
       p++;
   }                                           95cf
}                                              95cf
/*int wa[maxn],wb[maxn],wv[maxn],ws[maxn       8a3e
  ];
int cmp(int *r,int a,int b,int l)              3aa5
{return r[a]==r[b]&&r[a+l]==r[b+l];}           9692
void da(int *r,int *sa,int n,int m)            cdcb
{                                              4506
  int i,j,p,*x=wa,*y=wb,*t;                    8fa8
  for(i=0;i<m;i++) ws[i]=0;                    4f37
  for(i=0;i<n;i++) ws[x[i]=r[i]]++;            960c
  for(i=1;i<m;i++) ws[i]+=ws[i–1];             be11
  for(i=n–1;i>=0;i––) sa[––ws[x[i]]]=i;        5f6d
  for(j=1,p=1;p<n;j*=2,m=p)                    efbc
  {                                            4506
    for(p=0,i=n–j;i<n;i++) y[p++]=i;           5b7b
    for(i=0;i<n;i++) if(sa[i]>=j) y[p++]=      dc1c
      sa[i]–j;
    for(i=0;i<n;i++) wv[i]=x[y[i]];            792f
    for(i=0;i<m;i++) ws[i]=0;                  4f37
7f78     for(i=0;i<n;i++) ws[wv[i]]++;
be11     for(i=1;i<m;i++) ws[i]+=ws[i–1];
f3a2     for(i=n–1;i>=0;i——) sa[——ws[wv[i]]]=y[i];
0b94     for(t=x,x=y,y=t,p=1,x[sa[0]]=0,i=1;i<n;i++)
fef3     x[sa[i]]=cmp(y,sa[i–1],sa[i],j)?p–1:p++;
95cf   }
fe38 }*/以下是
427e //算法DC3  O(n)
427e //r 数组和sa 数组的大小都要是3*n下面的三行都是必须的。。为了避免名冲突
427e //
427e //#define F(x) ((x)/3+((x)%3==1?0:tb))
427e //#define G(x) ((x)<tb?(x)*3+1:((x)–tb)*3+2)
427e //int wa[maxn],wb[maxn],wv[maxn],ws_[maxn];
9750 int c0(int *r,int a,int b)
4506 {
d7f5   return r[a]==r[b]&&r[a+1]==r[b+1]&&r[a+2]==r[b+2];
95cf }
6829 int c12(int k,int *r,int a,int b)
1258 {  if(k==2) return r[a]<r[b]||r[a]==r[b]&&c12(1,r,a+1,b+1);
8b9e    else return r[a]<r[b]||r[a]==r[b]&&wv[a+1]<wv[b+1];
95cf }
8df6 void sort_sa(int *r,int *a,int *b,int n,int m)
4506 {
a0f7    int i;
0fed    for(i=0;i<n;i++) wv[i]=r[a[i]];
c934    for(i=0;i<m;i++) ws_[i]=0;
f297    for(i=0;i<n;i++) ws_[wv[i]]++;
2d9a    for(i=1;i<m;i++) ws_[i]+=ws_[i–1];
1285    for(i=n–1;i>=0;i——) b[——ws_[wv[i]]]=a[i];
4f2d    return;
95cf }
81ce void dc3(int *r,int *sa,int n,int m)
4506 {
455d    int i,j,*rn=r+n,*san=sa+n,ta=0,tb=(n+1)/3,tbc=0,p;
825e    r[n]=r[n+1]=0;
b6f1    for(i=0;i<n;i++) if(i%3!=0) wa[tbc++]=i;
427e
7371    sort_sa(r+2,wa,wb,tbc,m);
06a6    sort_sa(r+1,wb,wa,tbc,m);
79ea    sort_sa(r,wa,wb,tbc,m);
```

```
                                                              427e
    for(p=1,rn[F(wb[0])]=0,i=1;i<tbc;i++)                     64aa
    rn[F(wb[i])]=c0(r,wb[i–1],wb[i])?p–1:p++;                 16b5
                                                              427e
    if(p<tbc) dc3(rn,san,tbc,p);                              778a
    else for(i=0;i<tbc;i++) san[rn[i]]=i;                     6879
                                                              427e
    for(i=0;i<tbc;i++) if(san[i]<tb) wb[ta++]=san[i]*3;       37b9
                                                              427e
    if(n%3==1) wb[ta++]=n–1;                                  6cbc
                                                              427e
    sort_sa(r,wb,wa,ta,m);                                    0fde
                                                              427e
    for(i=0;i<tbc;i++) wv[wb[i]=G(san[i])]=i;                 e658
    for(i=0,j=0,p=0;i<ta && j<tbc;p++)                        79d3
                                                              427e
    sa[p]=c12(wb[j]%3,r,wa[i],wb[j])?wa[i++]:wb[j++];         1a17
                                                              427e
    for(;i<ta;p++) sa[p]=wa[i++];                             d05f
    for(;j<tbc;p++) sa[p]=wb[j++];                            b151
                                                              427e
}                                                             95cf
                                                              427e
//height 数组：定义height[i]=suffix(sa[i–1])和suffix(sa[i])的最长公共前缀算法如下，复杂度   427e
//（）On一个重要的结论：                                       427e
//suffix(j) 和suffix(k) 的最长公共前缀为height[rank[j]+1],     427e
//height[rank[j]+2], height[rank[j]+3], … ,height[rank[k中的最小值。]]待排序的字符串放在   427e
                                                              427e
//r 数组中，从r到[0]r[n，长度为–1]，且最大值小n于               427e
//。为了函数操作的方便，约定除mr[n外所有的–1]r[i都大于]0, r[n。–1]=0函数结束后，结果放在   427e
//sa 数组中，从sa到[0]sa[n。–1]                                427e
int rank[maxn],height[maxn];                                  8039
void calheight(int *r,int *sa,int n)                          4772
{                                                             4506
    int i,j,k=0;                                              ff68
    for(i=1;i<=n;i++) rank[sa[i]]=i;                          3748
    for(i=0;i<n;height[rank[i++]]=k)                          d007
    for(k?k——:0,j=sa[rank[i]–1];r[i+k]==r[j+k];k++);          3e0a
    return;                                                   4f2d
}                                                             95cf
```

```
427e
4fe3 char str[100];
5731 int  r[100];
01f2 int  sa[100];
299c int main()
4506 {
582c   while(cin>>str){
8d4d     int L=strlen(str);
e2ff     for(int i=0;i<L;i++)r[i]=str[i]–'0';
f411     for(int i=0;i<L;i++)sa[i]=i;
8426     r[L]=0;
f76f     da(r,sa,L+1,200);
7bae     cout<<"haha\n";
790a     for(int i=0;i<L;i++)
0c37       cout<<sa[i]<<' ';cout<<endl;
5271     calheight(r,sa,L);
790a     for(int i=0;i<L;i++)
8560       cout<<height[i]<<' ';cout<<endl;
95cf   }
95cf }
```

5.5 线段树区间修改

```
59b9 #include <cstdio>
427e
5963 struct IT
4506 {
ff6f   #define MN 1048577*2
d788   long long F[MN],D[MN];
d7af   int N;
651e   long long C,Ans;
ecbf   int ll,rr;
f365   void upd( int k, int l, int r)
4506   {
2d0d     if (D[k])
4506     {
1111       F[k]+=D[k]*((long long)r–l+1);
1f3f       if (k<N) D[k<<1]+=D[k],D[(k<<1)
             +1]+=D[k];
eef5       D[k]=0;
95cf     }
95cf   }
69ea   void find( int k, int l, int r)
4506   {
eed2     upd(k,l,r);
30f7     if (r<ll||l>rr) return ;
456c     if (ll<=l&&rr>=r)
4506     {
a619       Ans+=F[k];
4f2d       return ;
95cf     }
8f58     find(k<<1,l,(l+r)>>1);
b2c9     find((k<<1)+1,((l+r)>>1)+1,r);
95cf   }
  void Add( int k, int l, int r)                       25be
  {                                                    4506
    upd(k,l,r);                                        eed2
    if (r<ll||l>rr) return ;                           30f7
    if (ll<=l&&rr>=r)                                  456c
    {                                                  4506
      D[k]+=C;                                         f1c1
      upd(k,l,r);                                      eed2
      return ;                                         4f2d
    }                                                  95cf
    Add(k<<1,l,(l+r)>>1);                              817a
    Add((k<<1)+1,((l+r)>>1)+1,r);                      3afb
    F[k]=F[k<<1]+F[(k<<1)+1];                          7514
  }                                                    95cf
  void init( int NN)                                   4d16
  {                                                    4506
    for (N=1;N<NN;N<<=1);                              2944
    for (int i=1;i<=NN;i++) scanf("%I64d               b3f6
      ",&F[i+N–1]);
    for (int i=N–1;i;i––) F[i]=F[i<<1]+F               7991
      [(i<<1)+1];
  }                                                    95cf
  long long search( int _l, int _r)                    a363
  {                                                    4506
    ll=_l;rr=_r;                                       a0c0
    Ans=0;                                             ea34
    find(1,1,N);                                       cc62
    return Ans;                                        ba71
  }                                                    95cf
  void add( int _l, int _r, long long c)               893e
  {                                                    4506
    ll=_l;rr=_r;C=c;                                   62cd
    Add(1,1,N);                                        31fd
  }                                                    95cf
} T;                                                   eb81
                                                       427e
int N,M,a,b,c;                                         12d2
                                                       427e
int main()                                             299c
{                                                      4506
  scanf("%d%d",&N,&M);                                 a82c
  T.init(N);                                           d49b
  for (int i=1;i<=M;i++)                               a874
  {                                                    4506
    scanf("%d%d%d",&a,&b,&c);                          677b
    T.Add(a,b,c);                                      3069
    printf("%lld\n",T.search(i,N));                    c396
  }                                                    95cf
}                                                      95cf
```

5.6 AhoCorasick

```
#include<iostream>                                     e0a5
#include<cstring>                                      ef2f
```

```
59b9 #include<cstdio>
c928 #include<cmath>
09f7 #include<vector>
acb9 #include<queue>
54ff #include<algorithm>
421c using namespace std;
427e //const int inf=INT_MAX;
427e //int maxInt   =   0x7FFFFFFF;//   32 bit
427e //long  maxLong  =   0x7FFFFFFFFFFFFFFF;  // 64   bit
c115 #define ll long long
5841 const int inf= 0x7FFFFFFF;
b43a const char atcg[]="ATCG";
94ec const int kind=4;
35b8 int n,m;
586b char in[105];
cbdf struct ahocorasick{
5520   static const int undef=0;//初始化tag
df17   static const int maxn=1<<10;
8c42   static const int charset子节点个数=4;//
4022   int end;
2e34   int tag[maxn],fail[maxn],trie[maxn][charset];
5d53   void init(){
59e7     tag[0]=undef;
196d     fill(trie[0],trie[0]+charset,-1);
feb8     end=1;
95cf   }
f53c   void add(int m,const char *s,int t)插入状态为{//t
ff1e     int p=0,index;
356f     for(int i=0;i<m;i++){
c273       index=strchr(atcg,s[i])-atcg;
9c09       if(trie[p][index]==-1){
5e5e         tag[end]=undef;
f543         fill(trie[end],trie[end]+charset,-1);
bc80         trie[p][index]=end++;
95cf       }
79dc       p=trie[p][index];
95cf     }
7b5e     tag[p]|=t;
95cf   }
2114   void build(){
dfc8     queue<int>bfs;
a7a6     fail[0]=0;
6830     for(int i=0;i<charset;i++){
131c       if(trie[0][i]!=-1){
9b4d         fail[trie[0][i]]=0;
79f5         bfs.push(trie[0][i]);
95cf       }
649a       else
```

```
        trie[0][i]=0;                              6c43
      }                                            95cf
      while(!bfs.empty()){                         88bb
        int p=bfs.front();                         e42e
        tag[p]|=tag[fail[p]];                      38ff
        bfs.pop();                                 1a76
        for(int i=0;i<charset;i++){                6830
          if(trie[p][i]!=-1){                      9f81
            fail[trie[p][i]]=trie[fail[p]][i];     076e
            bfs.push(trie[p][i]);                  659d
          }                                        95cf
          else                                     649a
            trie[p][i]=trie[fail[p]][i];           7720
        }                                          95cf
      }                                            95cf
    }                                              95cf
}ac;                                               0244
bool dp[2][1<<11][1<<11];                          7af7
int sum[1<<11];                                    1abe
int w[101];                                        dfbc
int main(){                                        3117
  int i,j,k,t,cur,pre;                             8b27
  while(scanf("%d%d",&n,&m)+1){                    178e
    ac.init();                                     c5bc
    for(i=0;i<n;i++){                              ee09
      scanf("%s %d",in,&w[i]);                     137a
      ac.add(strlen(in),in,1<<i);                  aa86
    }                                              95cf
    ac.build();                                    b345
    memset(sum,0,sizeof(sum));                     7a1a
    for(i=0;i<(1<<10);i++)                         bbfc
    for(j=0;j<n;j++){                              9e25
      if(i&(1<<j))                                 5f0a
      sum[i]+=w[j];                                c3c1
    }                                              95cf
    memset(dp,false,sizeof(dp));                   3663
    cur=0,pre=1;                                   4b53
    dp[cur][0][0]=true;                            8fc2
    for(i=0;i<m;i++){                              494a
      swap(cur,pre);                               f70a
      memset(dp[cur],false,sizeof(dp[cur]));       f281
      for(int j=0;j<ac.end;j++)                    3ddf
        for(k=0;k<4;k++){                          24ec
          int nxt=ac.trie[j][k];                   9e0a
          for(t=0;t<(1<<n);t++)                    f4b4
          dp[cur][nxt][t|ac.tag[nxt]] |= dp[pre][j][t];   f9e8
        }                                          95cf
    }                                              95cf
    int ans=-1;                                    1aea
    for(i=0;i<(1<<n);i++)                          4d5d
```

```
5ad9       for(j=0;j<ac.end;j++){
1daa         if(dp[cur][j][i]){
7968           ans=max(ans,sum[i]);
95cf         }
95cf       }
3214     if(ans<0)puts("No Rabbit after
           2012!");
e0fa     else cout<<ans<<endl;
95cf   }
95cf }
```

5.7 kmp

```
e0a5 #include<iostream>
59b9 #include<cstdio>
8c52 #include<map>
6326 #include<set>
ef2f #include<cstring>
2349 #include<string>
421c using namespace std;
427e 
427e //extended kmp
427e //nxt[] & ext[] should be clarify out of
       the function
99b8 void ExtendKmp(char s[],int ls,char t[],
       int lt)
4506 {
c8ed   int i,j,k;
3f1e   int Len,L;
427e 
27ef   j=0;
8a9f   while(t[j+1]==t[j]&&j+1<lt) j++;
6a3e   nxt[1]=j,k=1;
427e 
7588   for(i=2;i<lt;i++){
c163     Len=k+nxt[k],L=nxt[i-k];
0c66     if(Len>L+i) nxt[i]=L;
037f     else{
00d5       j=Len-i>0?Len-i:0;
e0db       while(t[i+j]==t[j]&&i+j<lt) j++;
a782       nxt[i]=j,k=i;
95cf     }
95cf   }
27ef   j=0;
70fe   while(s[j]==t[j]&&j<lt&&j<ls) j++;
bd73   ext[0]=j,k=0;
427e 
bf8b   for(i=1;i<ls;i++){
8241     Len=k+ext[k],L=nxt[i-k];
93ba     if(Len>L+i) ext[i]=L;
037f     else{
00d5        j=Len-i>0?Len-i:0;
d6e6        while(s[i+j]==t[j]&&i+j<ls&&j<lt)
              j++;
```

```
       ext[i]=j,k=i;                        8df1
    }                                       95cf
  }                                         95cf
}                                           95cf
                                            427e
//kmp                                       427e
                                            427e
void get_ne(char* p,int *nex)               2ab8
{                                           4506
     int i=0;int j=-1;                      6b27
     nex[0]=-1;                             9d06
     int L=strlen(p);                       26de
     while(i<L)                             f044
     {                                      4506
         if(j==-1||p[i]==p[j])              84ee
         {                                  4506
           i++;j++;                         b209
           nex[i]=j;                        f023
         }                                  95cf
         else                               649a
           j=nex[j];                        8051
                                            427e
     }                                      95cf
}                                           95cf
```

5.8 Palindromic

```
void manacher( int* r, int len, int* p )    cce1
{                                           4506
  int i, id, right = 0;                     cc69
  for( i = 0; i < len; ++i )                ab12
  {                                         4506
    if( i < right )                         27c2
      p[i] = min(p[2*id-i], right-i);       3fb4
    else                                    649a
      p[i] = 1;                             0d63
    while( i+p[i] < len && i-p[i] >= 0      357d
      && r[i+p[i]] == r[i-p[i]] )           3a1f
      p[i]++;                               0eac
    if( p[i] + i > right )                  5bba
    {                                       4506
      right = p[i]+i;                       d237
      id = i;                               d14a
    }                                       95cf
  }                                         95cf
}                                           95cf
```

6 图论

6.1 Hamilton

```
//Hamilton 回路存在条件:                     427e
// 每个点的度数超过一半点数                   427e
```

```
427e  // 复杂度 N^3 左右
427e  //@
59b9  #include <cstdio>
ef2f  #include <cstring>
427e
4c18  #define MN 201
427e
b7b3  int map[MN][MN];
aa15  int N,M;
53a5  int ans[MN];
427e
90a3  void reverse( int ans[MN], int s, int t)
4506  {
e87d    int temp;
d7ae    while (s<t)
4506    {
cb13      temp=ans[s];
9ace      ans[s]=ans[t];
8ba6      ans[t]=temp;
3c90      s++;t--;
95cf    }
95cf  }
427e
4871  void Hamilton()
4506  {
f392    int s=1,t;
6d29    int ansi=2,i,j,w,temp;
3fe0    bool vis[MN]={false};
87fb    memset(vis,0,sizeof vis);
7790    for (i=1;i<=N;i++)
0ba4      if (map[s][i]) break;
5093    t=i;
6299    vis[s]=vis[t]=1;
ea3a    ans[0]=s;ans[1]=t;
1f75    while (1)
4506    {
1f75      while (1)
4506      {
7790        for (i=1;i<=N;i++)
44dc          if (map[t][i]&&!vis[i])
4506          {
4d1d            ans[ansi++]=i;
1080            vis[i]=1;
5093            t=i;
6173            break;
95cf          }
882c        if (i>N) break;
95cf      }
4fa4      w=ansi-1;
14dc      i=0;
8ed9      reverse(ans,i,w);
02b3      temp=s;
f770      s=t;t=temp;
```

```
    while (1)                                          1f75
    {                                                  4506
      for (i=1;i<=N;i++)                               7790
        if (map[t][i]&&!vis[i])                        44dc
        {                                              4506
          ans[ansi++]=i;vis[i]=1;                      5260
          t=i;                                         5093
          break;                                       6173
        }                                              95cf
      if (i>N) break;                                  882c
    }                                                  95cf
    if (!map[s][t])                                    fd5b
    {                                                  4506
      for (i=1;i<ansi-2;i++)                           110b
        if (map[ans[i]][t]&&map[s][ans[i               2fc4
          +1]]) break;
      w=ansi-1;                                        4fa4
      i++;                                             a42b
      t=ans[i];                                        642b
      reverse(ans,i,w);                                8ed9
    }                                                  95cf
    if (ansi==N) return;                               34fc
    for (j=1;j<=N;j++)                                 5cf5
    {                                                  4506
      if (vis[j]) continue;                            ccf8
      for (i=1;i<ansi-2;i++)                           110b
        if (map[ans[i]][j]) break;                     2d6a
      if (map[ans[i]][j]) break;                       2d6a
    }                                                  95cf
    s=ans[i-1];                                        7dd0
    t=j;                                               9f0e
    reverse(ans,0,i-1);                                905e
    reverse(ans,i,ansi-1);                             b398
    ans[ansi++]=j;                                     be20
    vis[j]=1;                                          59bb
  }                                                    95cf
}                                                      95cf
                                                       427e
int main()                                             299c
{                                                      4506
  while (scanf("%d%d",&N,&M)!=EOF)                     60b7
  {                                                    4506
    memset(map,0,sizeof map);                          acaf
    int a,b;                                           e635
    while (M--)                                        3e4a
    {                                                  4506
      scanf("%d%d",&a,&b);                             a6b8
      map[a][b]=map[b][a]=1;                           69bc
    }                                                  95cf
                                                       427e
    Hamilton();                                        16db
    for (int i=0;i<N-1;i++) printf("%d ",              ea0f
      ans[i]);
```

```
8c45      printf("%d\n",ans[N-1]);
07ea      for (int i=0;i<N-1;i++)
6a10        if (!map[ans[i]][ans[(i+1)%N]])
              while (1);
95cf    }
7021    return 0;
95cf  }
```

6.2 HopcraftKarp

```
427e  // HK 解二分图匹配
427e  // 复杂度 O(sqrt(N)*M)
427e  //*@
59b9  #include <cstdio>
ef2f  #include <cstring>
427e
4cbf  #define MN 50001
0889  #define MM 150001
427e
427e
673f  struct E_Node
4506  {
28a5    int v,ne;
f9e1  } E[MM];
427e
7c0e  int R[MN],e_sz;
aa15  int N,M;
427e
d699  void Add( int a, int b)
4506  {
ddad    E[++e_sz].v=b;E[e_sz].ne=R[a];R[a]=e_sz
          ;
95cf  }
427e
88f1  void init()
4506  {
1282    e_sz=0;
ab17    memset(R,0,sizeof R);
95cf  }
427e
427e
8a67  int ans,qsz;
427e
8ec5  int Qu[MN*2],Lx[MN],Ly[MN],Cx[MN],Cy[MN];
427e
e459  bool BFS()
4506  {
c93c    int p,u,v;
f25a    bool flag=0;
1034    qsz=0;
c802    for (int i=1;i<=N;i++)
a0d0      if (Cx[i]==-1) Qu[++qsz]=i,Lx[i]=1;
3fd6    memset(Ly,0xff,sizeof(Ly[0])*(M+1));
aa16    for (int i=1;i<=qsz;i++)
```

```
  {                                             4506
    u=Qu[i];                                    1a30
    for (p=R[u];p;p=E[p].ne)                    f05c
    {                                           4506
      v=E[p].v;                                 43a3
      if (Ly[v]==-1)                            dc20
      {                                         4506
        Ly[v]=Lx[u]+1;                          a04e
        if (Cy[v]==-1) flag=1;                  0d23
        else                                    649a
        {                                       4506
          Qu[++qsz]=Cy[v];                      19cf
          Lx[Cy[v]]=Ly[v]+1;                    a158
        }                                       95cf
      }                                         95cf
      }                                         95cf
  }                                             95cf
  return flag;                                  01d8
}                                               95cf
                                                427e
bool Dfs( int k)                                ec04
{                                               4506
  for (int p=R[k];p;p=E[p].ne)                  0470
    if (Ly[E[p].v]==Lx[k]+1)                    68e3
    {                                           4506
      Ly[E[p].v]=-1;                            1ef9
      if (Cy[E[p].v]==-1||Dfs(Cy[E[p].v])       7c7e
        )
      {                                         4506
        Cy[E[p].v]=k;                           f491
        Cx[k]=E[p].v;                           e929
        return 1;                               7459
      }                                         95cf
    }                                           95cf
  return 0;                                     7021
}                                               95cf
                                                427e
                                                427e
      Add(a,b);                                 7b71
                                                427e
    ans=0;                                      7360
    memset(Cx,0xff,(N+1)*sizeof(Cx[0]));        f3ef
    memset(Cy,0xff,(M+1)*(sizeof(Cy[0])))       4390
      ;
    while (BFS())                               6ab9
    {                                           4506
      for (int i=N;i;i--)                       3c73
        if (Cx[i]==-1&&Dfs(i)) ans++;           3ed3
    }                                           95cf
    printf("%d\n",ans);                         53b1
}                                               95cf
```

6.3 HopcraftKarp

```
427e  // HK 解二分图匹配复杂度
427e  // O(sqrt(N)*M)
427e  //*@
59b9  #include <cstdio>
ef2f  #include <cstring>
427e
4cbf  #define MN 50001
0889  #define MM 150001
427e
427e
673f  struct E_Node
4506  {
28a5    int v,ne;
f9e1  } E[MM];
427e
7c0e  int R[MN],e_sz;
aa15  int N,M;
427e
d699  void Add( int a, int b)
4506  {
ddad    E[++e_sz].v=b;E[e_sz].ne=R[a];R[a]=e_sz
          ;
95cf  }
427e
88f1  void init()
4506  {
1282    e_sz=0;
ab17    memset(R,0,sizeof R);
95cf  }
427e
427e
8a67  int ans,qsz;
427e
8ec5  int Qu[MN*2],Lx[MN],Ly[MN],Cx[MN],Cy[MN];
427e
e459  bool BFS()
4506  {
c93c    int p,u,v;
f25a    bool flag=0;
1034    qsz=0;
c802    for (int i=1;i<=N;i++)
a0d0      if (Cx[i]==–1) Qu[++qsz]=i,Lx[i]=1;
3fd6    memset(Ly,0xff,sizeof(Ly[0])*(M+1));
aa16    for (int i=1;i<=qsz;i++)
4506    {
1a30      u=Qu[i];
f05c      for (p=R[u];p;p=E[p].ne)
4506      {
43a3        v=E[p].v;
dc20        if (Ly[v]==–1)
4506        {
a04e          Ly[v]=Lx[u]+1;
0d23          if (Cy[v]==–1) flag=1;
649a          else
4506          {
19cf            Qu[++qsz]=Cy[v];
a158            Lx[Cy[v]]=Ly[v]+1;
95cf          }
95cf        }
95cf        }
95cf    }
01d8    return flag;
95cf  }
427e
ec04  bool Dfs( int k)
4506  {
0470    for (int p=R[k];p;p=E[p].ne)
68e3      if (Ly[E[p].v]==Lx[k]+1)
4506      {
1ef9        Ly[E[p].v]=–1;
7c7e        if (Cy[E[p].v]==–1||Dfs(Cy[E[p].v])
            )
4506        {
f491          Cy[E[p].v]=k;
e929          Cx[k]=E[p].v;
7459          return 1;
95cf        }
95cf      }
7021    return 0;
95cf  }
427e
427e
7b71        Add(a,b);
427e
7360      ans=0;
f3ef      memset(Cx,0xff,(N+1)*sizeof(Cx[0]));
4390      memset(Cy,0xff,(M+1)*(sizeof(Cy[0])))
            ;
6ab9      while (BFS())
4506      {
3c73        for (int i=N;i;i—)
3ed3          if (Cx[i]==–1&&Dfs(i)) ans++;
95cf      }
53b1      printf("%d\n",ans);
95cf  }
```

6.4 Hungary

```
427e  // 匈牙利算法解二分图匹配
427e  // 一般小于 NM
427e  //*@
e219  bool Vis[MN];
bf70  int Link[MN];
2a22  int Cache[MN];
2fd8  int Csz;
427e
b96a  bool find( int k)
```

```
4506 {
0470   for (int p=R[k];p;p=E[p].ne)
4d33     if (!Vis[E[p].v])
4506     {
39cb       Vis[E[p].v]=1;
6bbc       Cache[++Csz]=E[p].v;
f267       if (!Link[E[p].v])
4506       {
17ae         Link[E[p].v]=k;
3361         return true;
95cf       }
fc61       if (find(Link[E[p].v]))
4506       {
17ae         Link[E[p].v]=k;
3361         return true;
95cf       }
95cf     }
438e   return false;
95cf }
427e
f531     memset(Link,0,sizeof Link);
c802     for (int i=1;i<=N;i++)
4506     {
73c9       for (int k=1;k<=Csz;k++)
c926         Vis[Cache[k]]=0;
19e8         Csz=0;
8f54         if (find(cc(i,j))) ans++;
95cf     }
```

6.5 KM

```
427e //KM
427e // 二分图最佳匹配
427e // 点数不等时添加虚拟点，与所有点的边权为 0
427e // 点数不等时不能将边权变负，要用 INF 去减
427e //N^3
427e
59b9 #include <cstdio>
ef2f #include <cstring>
427e
1d4b #define MN 301
1cc6 #define INF 0x7fffffff
427e
d7e9 int w[MN][MN];
5897 int lx[MN],ly[MN];
d96c int linky[MN];
9859 int visx[MN],visy[MN];
9012 int slack[MN];
d7af int N;
6097 bool find(int x)
4506 {
28d0   visx[x] = true;
5647   for(int y = 1; y <=N; y++)
4506   {
```

```
    if(visy[y])                              aab6
      continue;                              b333
    int t = lx[x] + ly[y] - w[x][y];         44b7
    if(t==0)                                 d790
    {                                        4506
      visy[y] = true;                        d2aa
      if(linky[y]==-1 || find(linky[y]))     6def
      {                                      4506
        linky[y] = x;                        b930
        return true;                         3361
      }                                      95cf
    }                                        95cf
    else if(slack[y] > t)                    cf8b
      slack[y] = t;                          52e5
  }                                          95cf
  return false;                              438e
}                                            95cf
                                             427e
int KM()                                     45ef
{                                            4506
  int i,j;                                   576f
                                             427e
  memset(linky,-1,sizeof(linky));            76e3
  memset(ly,0,sizeof(ly));                   b7b2
  for(i = 1; i <=N; i++)                     7790
  {                                          4506
    lx[i] = -INF;                            7b3c
     for(j = 1; j <=N; j++)                  5cf5
      if(w[i][j] > lx[i])                    7f0c
        lx[i] = w[i][j];                     c66c
  }                                          95cf
                                             427e
  for(int x = 1; x <=N; x++)                 67de
  {                                          4506
    for(i = 1; i <=N; i++)                   7790
      slack[i] = INF;                        a6e7
    while(true)                              66e0
    {                                        4506
      memset(visx,0,sizeof(visx));           e77f
      memset(visy,0,sizeof(visy));           688c
      if(find(x))                            3333
        break;                               6173
      int d = INF;                           065a
      for(i = 1; i <=N; i++)                 7790
      {                                      4506
        if(!visy[i] && d > slack[i])         df80
          d = slack[i];                      bae8
      }                                      95cf
      for(i = 1; i <=N; i++)                 7790
      {                                      4506
        if(visx[i])                          f248
          lx[i] -= d;                        529c
      }                                      95cf
      for(i = 1; i <=N; i++)
      {
        if(visy[i])
           ly[i] += d;
        else
          slack[i] -= d;
      }
    }
  }
  int result = 0;
  for(i = 1; i <=N; i++)
  if(linky[i]>-1)
    result += w[linky[i]][i];
  return result;
}

int main()
{
  while (scanf("%d",&N)!=EOF)
  {
    memset(w,0,sizeof w);
    for (int i=1;i<=N;i++)
      for (int j=1;j<=N;j++) scanf("%d",&
        w[i][j]);
    printf("%d\n",KM());
  }
}
```

6.6 lca

```
// 建树和倍增求 lca

  //buildtree

  int dep[MN],root[MN];
  int ancestor[MN][logMN];
  void buildtree( int k)
  {
    vis[k]=1;
    for (int p=R[k];p;p=E[p].ne)
    {
      if (vis[E[p].v]) continue;
      ancestor[E[p].v][0]=k,dep[E[p].v]=
        dep[k]+1,root[E[p].v]=root[k],
        buildtree(E[p].v);
    }
  }
  //lca_init
  // 可以处理森林
  void lca_init( int N)
  {
    memset(ancestor,0,sizeof ancestor);
    for (int i=1;i<=N;i++) vis[i]=0;
    for (int i=1;i<=N;i++)
      if (ancestor[i][0]==0) dep[i]=1,
        root[i]=i,buildtree(i);

    for (int k=1;k<=log2(N);k++)
      for (int i=1;i<=N;i++)
        ancestor[i][k]=ancestor[ancestor[
          i][k-1]][k-1];
  }

  //return lca(a,b)
  //return -1 if a,b in diffirent tree
  int lca_query( int a, int b, int N)
  {
    int ans=dep[a]+dep[b];
    if (root[a]!=root[b]) return -1;
    if (a==b) return a;
    if (dep[a]>dep[b]) swap(a,b);
    for (int k=0;k<=log2(N);k++) if ((1<<
      k)&delta) b=ancestor[b][k];
    if (a==b) return a;
    for (int k=log2(N);k>=0;k--) if (
      ancestor[a][k]!=ancestor[b][k]) a=
      ancestor[a][k],b=ancestor[b][k];
    return ancestor[a][0];
  }
```

6.7 MaxcostMaxflow

```
// 最小费用路增广 bySPFA
//*@
#include <cstdio>
#include <cstring>
#include <queue>

using namespace std;

#define MN 30
#define MM 100000
#define INF 0x7fffffff

int T,N,M,K;
int Lim[MN],Like[MN][MN],l,cases;

struct MaxcostMaxflow
{
  struct E_Node
  {
    int v,f,c,op,ne;
  } E[MM];
  int R[MN],sz;
  int S,T;
  void Add( int a, int b, int c, int d)
  {
    E[++sz].v=b;E[sz].ne=R[a];R[a]=sz;E[
          sz].f=c;E[sz].c=d;
95cf  }
d3c6  void Ins( int a, int b, int c, int d)
4506  {
4238    if (!c) return ;
18b4    Add(a,b,c,d);E[sz].op=sz+1;
11e8    Add(b,a,0,-d);E[sz].op=sz-1;
95cf  }
7d24  bool B[MN];
0e46  int D[MN],F[MN];
3d1d  int maxcost,maxflow;
88f1  void init()
4506  {
1bb9    sz=0;
ab17    memset(R,0,sizeof R);
1fd4    maxcost=maxflow=0;
95cf  }
aafa  queue<int> Q;
64de  bool SPFA()
4506  {
3ccf    int u,m,p;
a058    while (!Q.empty()) Q.pop();
e753    Q.push(S);
5480    for (int i=1;i<=T;i++) D[i]=-INF,B[i
          ]=0;
eb5f    D[S]=0;B[S]=1;
1b18    while (!Q.empty())
4506    {
50ae      u=Q.front();
f2f8      Q.pop();
f05c      for (p=R[u];p;p=E[p].ne)
94ec        if (E[p].f&&D[u]+E[p].c>D[E[p].v
              ])
4506        {
32ff          D[E[p].v]=D[u]+E[p].c;
6c54          F[E[p].v]=p;
d185          if (!B[E[p].v])
4506          {
fa94            B[E[p].v]=1;
bae3            if (!Q.empty()&&D[Q.front()]<
                  D[E[p].v])
4506            {
8b65              Q.push(Q.front());
99c8              Q.front()=E[p].v;
887e            } else Q.push(E[p].v);
95cf          }
95cf        }
fa2d      B[u]=0;
95cf    }
d2b6    if (D[T]==-INF) return 0;
649a    else
4506    {
c088      u=T;m=INF;
```

```
      while (u!=S)                          870d
      {                                     4506
        m=min(m,E[F[u]].f);                 f593
        u=E[E[F[u]].op].v;                  ace4
      }                                     95cf
      maxflow+=m;                           d1e9
      maxcost+=m*D[T];                      8902
      u=T;                                  3aaf
      while (u!=S)                          870d
      {                                     4506
        E[F[u]].f-=m;                       4712
        E[E[F[u]].op].f+=m;                 d9c0
        u=E[E[F[u]].op].v;                  ace4
      }                                     95cf
      return 1;                             7459
    }                                       95cf
  }                                         95cf
} G;                                        ff9a
/*                                          87e7
    G.S=N+M+1;G.T=G.S+1;                    664a
    for (int j=1;j<=M;j++) G.Ins(N+j,G.T    69b7
      ,1,0);
    while (G.SPFA());                       20d3
    if (G.maxcost+M-G.maxflow>=1) puts("    6bcf
      YES");
    else puts("NO");                        9418
*/                                          f2b5
```

6.8 NetworkFlow

```
//Dinic                                     427e
// 当前弧、多路增广、断层优化                  427e
//*@                                        427e
#include <cstring>                          ef2f
#define MN 1000                             45b5
#define MM 100000                           8c3b
                                            427e
struct E_Node                               673f
{                                           4506
  int a,b,opp,ne;                           6221
  int f;                                    e53b
} E[MM];                                    f9e1
                                            427e
int R[MN],e_sz;                             7c0e
                                            427e
                                            427e
                                            427e
struct s_node{                              a035
  int v,p;                                  99ad
} s[MN];                                    f95b
                                            427e
                                            427e
int top,N,M,S,T,ans;                        df6b
int Q[MN],L[MN];                            f90e
```

```
4b4a bool hash[MN];
427e
88f1 void init()
4506 {
ab17   memset(R,0,sizeof R);
1282   e_sz=0;
7360   ans=0;
95cf }
427e
0a45 void add( int a, int b, int c){
5242   E[++e_sz].a=a;E[e_sz].b=b;E[e_sz].f=c;
ae1c   E[e_sz].ne=R[a];R[a]=e_sz;
95cf }
427e
e9ba void ins( int a, int b, int c){
00d3   add(a,b,c);
c75f   E[e_sz].opp=e_sz+1;
131a   add(b,a,0);
f992   E[e_sz].opp=e_sz–1;
95cf }
427e
6193 int BFS(){
ad27   int h,t,p;
33f5   memset(L,0,sizeof(L));
879c   Q[h=t=1]=S;
a7f5   L[S]=1;
e2aa   for (;h<=t;h++){
837d     for (p=R[Q[h]];p;p=E[p].ne)
ca1d       if (E[p].f&&(!L[E[p].b])){
713f         L[E[p].b]=L[Q[h]]+1;
3c1d         Q[++t]=E[p].b;
95cf       }
9296     if (L[T]) return L[T];
95cf   }
e010   return L[T];
95cf }
427e
8cef int augmented(){
5dd6   int min=0x7fffffff;
eb8e   int mins;
fb9d   for (int i=1;i<top;i++)
9a8b     if (E[s[i].p].f<min){
6a1b       min=E[s[i].p].f;
7441       mins=i;
95cf     }
c66b   for (int i=1;i<top;i++){
9eb5     E[s[i].p].f–=min;
b362     E[E[s[i].p].opp].f+=min;
95cf   }
2294   ans+=min;
9407   return mins;
95cf }
427e
```

```
void Dinic(){                                    9690
  bool f;                                        2f55
  int *p,*v;                                     a591
  s[top=1].v=S;s[1].p=0;                         10a0
  memset(hash,true,sizeof(hash));                d9aa
  while (top){                                   55e4
    v=&s[top].v;                                 0cc3
    p=&s[top].p;                                 ac2c
    if (*v==T){                                  3605
      top=augmented();                           454a
      continue;                                  b333
    }                                            95cf
    if (*p)                                      607d
      *p=E[*p].ne;                               c70d
    else                                         649a
      *p=R[*v];                                  27e8
    f=false;                                     3b7e
    for (;*p;*p=E[*p].ne)                        9933
      if (hash[E[*p].b]&&E[*p].f&&L[*v           4eeb
        ]+1==L[E[*p].b]){
        s[++top].v=E[*p].b;                      b97a
        s[top].p=0;                              5fad
        top++;                                   6aed
        f=true;                                  69dc
        break;                                   6173
      }                                          95cf
    if (!f)                                      d954
      hash[s[top].v]=false;                      cdee
    top––;                                       e9a7
  }                                              95cf
}                                                95cf
                                                 427e
//init();                                        427e
//ins(a,b,c);                                    427e
//while (BFS()) Dinic();                         427e
```

6.9 SCC

```
//SCC by Tarjan                                  427e
// nowc 连通分量个数                               427e
//*@                                             427e
#include<cstdio>                                 59b9
#include<cstring>                                ef2f
#include<stack>                                  8207
#include<algorithm>                              54ff
                                                 427e
using namespace std;                             421c
                                                 427e
#define MN 20001                                 ab66
#define MM 50001                                 b120
                                                 427e
int N,M,T;                                       fbb8
                                                 427e
struct Graph                                     0c98
```

```
4506 {
a981   struct E_Node{
28a5     int v,ne;
f9e1   } E[MM];
5586   int R[MN],sz;
0d5d   void Add( int a, int b){
1352     E[++sz].v=b;E[sz].ne=R[a];R[a]=sz;
95cf   }
870e   stack<int> S;
7d24   bool B[MN];
b156   int D[MN],L[MN],C[MN],SZ[MN];
85a3   int nowc,nowd;
9372   void Dfs( int k){
0c3d     D[k]=L[k]=++nowd;
d651     B[k]=1;S.push(k);
0470     for (int p=R[k];p;p=E[p].ne)
2f4c       if (B[E[p].v]) L[k]=min(L[k],D[E[p].v]);
8ee9       else if (!D[E[p].v]) Dfs(E[p].v),L[k]=min(L[k],L[E[p].v]);
35b1     if (D[k]==L[k]){
3b67       int v;
9c48       nowc++;
a69f       do{
a4a8         v=S.top();
1894         B[v]=0;
c9fd         SZ[nowc]++;
1039         S.pop();
9f0e         C[v]=nowc;
14a1       }while(v!=k);
95cf     }
95cf   }
6e8b   void Rebuild( Graph &GG)
4506   {
2776     GG.init();
c802     for (int i=1;i<=N;i++)
8220       for (int p=R[i];p;p=E[p].ne)
83af         if (C[E[p].v]!=C[i]) GG.Add(C[i],C[E[p].v]);
95cf   }
7a60   void SCC(){
c802     for (int i=1;i<=N;i++)
25b5       if (!D[i]) Dfs(i);
95cf   }
88f1   void init()
4506   {
80a6     nowc=nowd=0;
1bb9     sz=0;
a31c     memset(B,0,sizeof B);
ab17     memset(R,0,sizeof R);
d799     memset(D,0,sizeof D);
e11b     memset(SZ,0,sizeof SZ);
95cf   }
ff9a } G;
427e
427e //G.init();
427e //G.SCC();
427e //G.Rebuild(GG);
```

6.10 Vconnect

```
59b9 #include <cstdio>
ef2f #include <cstring>
54ff #include <algorithm>
6326 #include <set>
09f7 #include <vector>
8207 #include <stack>
c928 #include <cmath>
427e
421c using namespace std;
427e
6c7d #define MN 410000
3c56 #define MM 2000000
5eec #define logMN 19
682b #define pb(x) push_back(x)
427e
aa15 int N,M;
427e
0c98 struct Graph
4506 {
673f   struct E_Node
4506   {
a15a     int a,v,ne;
f9e1   } E[MM];
427e
7c0e   int R[MN],e_sz;
427e
427e
c5de   void Add( int a, int b, int c=0)
4506   {
e53f     if (a==b) return;
98b5     E[++e_sz].v=b;;E[e_sz].ne=R[a];R[a]=e_sz;E[e_sz].a=a;
95cf   }
427e
4c87   int SZ[MN];
80ca   int O[MM];
427e
88f1   void init()
4506   {
1282     e_sz=0;
ab17     memset(R,0,sizeof R);
427e
427e     //Edge_Connect
95cf   }
427e
427e
```

```
427e    //Rebuild重构图
427e    //
427e    //E[],R[],N,col[],color
427e    //G
060e    int _col[MN];
427e
6e8b    void Rebuild( Graph &GG)
4506    {
2776      GG.init();
f4f9      for (int i=1;i<=color;i++) GG.cut[i
            ]=0;
c802      for (int i=1;i<=N;i++)
4506      {
9f92        if (cut[i])
4506        {
79b7          GG.cut[col[i]=++color]=1;
8220          for (int p=R[i];p;p=E[p].ne)
2975            GG.Add(col[i],O[p]),GG.Add(O[p
              ],col[i]);
95cf        }
95cf      }
c5da      for (int i=1;i<=color;i++) GG.SZ[i
            ]=0;
5f3a      for (int i=1;i<=N;i++) GG.SZ[col[i
            ]]++;
95cf    }
427e
427e    //Edge_Connect边双连通分量
427e    //
427e    //E[],R[],N先去除重边
427e    //
427e
024e    bool vis[MN];
6e90    int dfs[MN],low[MN];
79ea    int col[MN];
9ba1    bool vise[MM];
57b2    int color,cc;
50d0    bool cut[MN];
ae27    stack<int> stk;
541b    vector<int> cedge;
427e
9276    int op( int p)
4506    {
29da      if (p&1) return p+1;
ddf2      else return p−1;
95cf    }
427e
b988  int ci,cp;
f0e2    void Cedge()
4506    {
a725      for (ci=0;ci<cedge.size();ci++)
4133        for (cp=R[cedge[ci]];cp;cp=E[cp].ne
          )
8245          if (col[cedge[ci]]==col[E[cp].v])
c7ad            O[cp]=O[op(cp)]=col[cedge[ci]];
95cf    }
427e
f311    int CO[MN];
325b  int pp[MN];
d821    void Dfs( int k)
4506    {
0b49      vis[k]=1;
4755      stk.push(k);
139f      low[k]=dfs[k]=++cc;
c399      CO[k]=0;
bf51  #define p pp[k]
3cb5      for (p=R[k];p;p=E[p].ne)
4506      {
044c        if (vise[p])
4506        {
427e  //          low[k]=min(low[k],dfs[E[p].v]);
b333          continue;
95cf        }
2847        vise[p]=vise[op(p)]=1;
5b6b        if (!vis[E[p].v]) Dfs(E[p].v),CO[k
              ]++,low[k]=min(low[k],low[E[p].v
              ]);
649a        else
4506        {
a333          low[k]=min(low[k],dfs[E[p].v]);
b333          continue;//
95cf        }
1d99        if (dfs[k]==1&&CO[k]>1) cut[k]=1;
4db3        else if (dfs[k]!=1&&dfs[k]<=low[E[p
              ].v]) cut[k]=1;
2020        if (dfs[k]<=low[E[p].v])
4506        {
427e
b786          cedge.clear();
410f          col[k]=++color;
ad56          col[E[p].v]=color;
0e36          cedge.pb(k);
f40f          cedge.pb(E[p].v);
9533          while (stk.top()!=E[p].v)
4506          {
3d2d            col[stk.top()]=color;
ffe2            cedge.pb(stk.top());
75b6            stk.pop();
95cf          }
75b6          stk.pop();
25dc          Cedge();
427e
427e        //stk.pop();
95cf        }
95cf      }
95cf    }
```

```
52c1 #undef p
427e
885f   void Edge_Connect()
4506   {
d235     memset(cut,0,sizeof cut);
77f1     while (stk.size()) stk.pop();
2707     for (int i=0;i<=e_sz;i++) vise[i]=0;
427e
1120     color=cc=0;
c802     for (int i=1;i<=N;i++)
e88f       vis[i]=0,col[i]=0;
c802     for (int i=1;i<=N;i++)
4506     {
395c       cc=0;
07c3       if (!vis[i]) Dfs(i);
56eb /*      if (stk.size())
4506       {
b704         ++color;
7099         while (stk.size())
4506         {
3d2d           col[stk.top()]=color;
75b6           stk.pop();
95cf         }
fe38       }*/
95cf     }
95cf   }
427e
427e   //buildtree建树
427e   //
727b   int dep[MN],root[MN];
0ae7   int ancestor[MN][logMN];
2a08   int F[MN][logMN];
320d   int W[MN][logMN];
f7d7   int P[MN];
bf51 #define p pp[k]
79ea   void buildtree( int k)
4506   {
0b49     vis[k]=1;
3cb5     for (p=R[k];p;p=E[p].ne)
4506     {
f90d       if (vis[E[p].v]) continue;
4522     /*  if (cut[k]&&cut[E[p].v])
4506       {
885d         puts("");
fe38       }*/
427e
3c65       if (cut[k]==1) P[E[p].v]++;
18f1       if (cut[E[p].v]==1) P[k]++;
0854       ancestor[E[p].v][0]=k,F[E[p].v][0]=
             SZ[E[p].v],dep[E[p].v]=dep[k]+1,
             root[E[p].v]=root[k],buildtree(E[
             p].v);
95cf     }
```

```
  }                                          95cf
#undef p                                     52c1
  //lca_init                                 427e
  //预处理lca可以处理森林                       427e
  //                                         427e
  void lca_init( int N)                      1b95
  {                                          4506
    memset(ancestor,0,sizeof ancestor);      5750
    for (int i=1;i<=N;i++) vis[i]=0,P[i      935d
      ]=0;
    for (int i=1;i<=N;i++)                   c802
      if (ancestor[i][0]==0) F[i][0]=SZ[i    be65
        ],dep[i]=1,root[i]=i,buildtree(i)
        ;
                                             427e
    for (int i=1;i<=N;i++)                   c802
    {                                        4506
      if (cut[i]) P[i]=0,F[i][0]=1;F[i       d611
        ][0]+=P[i];
    }                                        95cf
    for (int k=1;k<=log2(N);k++)             4a2e
      for (int i=1;i<=N;i++)                 c802
      {                                      4506
        ancestor[i][k]=ancestor[ancestor[    98fc
          i][k-1]][k-1];
        F[i][k]=F[i][k-1]+F[ancestor[i][k    0630
          -1]][k-1];
      }                                      95cf
  }                                          95cf
                                             427e
  //return lca(a,b)                          427e
  //return -1 if a,b in diffirent tree       427e
  int lca_query(int a, int b, int N)         82b0
  {                                          4506
    int ans=0,tans=0,tt=0;                   e66e
    if (root[a]!=root[b]) return 0;          53e6
    if (a==b) return F[a][0];                a9ad
    if (dep[a]>dep[b]) swap(a,b);            6371
    int delta=dep[b]-dep[a];                 d0b6
    ans=-delta;                              8c70
    for (int k=0;k<=log2(N);k++) if ((1<<    0abf
      k)&delta) ans+=F[b][k],b=ancestor[b
      ][k];
    if (a==b) return ans+F[a][0];            fdd8
    for (int k=log2(N);k>=0;k--) if (        8028
      ancestor[a][k]!=ancestor[b][k]) ans
      +=F[a][k]+F[b][k],ans-=1<<(k+1),a=
      ancestor[a][k],b=ancestor[b][k];
    return ans+F[a][0]+F[b][0]+F[ancestor    0c1e
      [a][0]][0]-2;
  }                                          95cf
                                             427e
} G,GG;                                      bef6
```

```
427e
427e
ea31 int cases;
427e
299c int main()
4506 {
e635   int a,b;
60b7   while (scanf("%d%d",&N,&M)!=EOF)
4506   {
1945     G.init();
427e     //Edge.clear();
a874     for (int i=1;i<=M;i++)
4506     {
a6b8       scanf("%d%d",&a,&b);
4bbc       a++;b++;
427e     //  if (Edge.find(make_pair(a,b))!=
           Edge.end()) continue;
03ff       G.Add(a,b);G.Add(b,a);
427e     //  Edge.insert(make_pair(a,b));
427e     //  Edge.insert(make_pair(b,a));
95cf     }
c01f     G.Edge_Connect();
b6c8     G.Rebuild(GG);
e9b6     GG.lca_init(G.color);
427e
9523     int T;
1fd9     scanf("%d",&T);
524a     printf("Case #%d:\n",++cases);
7d0e     while (T——)
4506     {
a6b8       scanf("%d%d",&a,&b);
4bbc       a++;b++;
442b       if (a==b) printf("%d\n",N–1);
eba2       else printf("%d\n",N–GG.lca_query(G
             .col[a],G.col[b],G.color));
95cf     }
885d     puts("");
95cf   }
95cf }
```

6.11 Stoer-Wagne

```
427e // 全局最小割
427e //N^3
427e //@
e0a5 #include <iostream>
59b9 #include <cstdio>
ef2f #include <cstring>
421c using namespace std;
47b3 const int maxn=510;
0541 int map[maxn][maxn];
5c83 int n;
c827 void contract(int x,int y)
4506 {
```

```
  int i,j;                                       576f
  for (i=0;i<n;i++)                              2dbf
    if (i!=x) map[x][i]+=map[y][i],map[i         0180
      ][x]+=map[i][y];
  for (i=y+1;i<n;i++) for (j=0;j<n;j++)          ffcb
  {                                              4506
    map[i–1][j]=map[i][j];                       ab88
    map[j][i–1]=map[j][i];                       c988
  }                                              95cf
  n——;                                           61b6
}                                                95cf
int w[maxn],c[maxn];                             78af
int sx,tx;                                       d7dd
int mincut()                                     9acc
{                                                4506
  int i,j,k,t;                                   74c3
  memset(c,0,sizeof(c));                         7dd1
  c[0]=1;                                        3bab
  for (i=0;i<n;i++) w[i]=map[0][i];              457c
  for (i=1;i+1<n;i++)                            a17f
  {                                              4506
    t=k=–1;                                      901d
    for (j=0;j<n;j++) if (c[j]==0&&w[j]>k        4d7b
      )
     k=w[t=j];                                   8647
    c[sx=t]=1;                                   d40b
    for (j=0;j<n;j++) w[j]+=map[t][j];           e5e3
  }                                              95cf
  for (i=0;i<n;i++) if (c[i]==0) return w        6bff
    [tx=i];
  }                                              95cf
int main()                                       299c
{                                                4506
  int i,j,k,m;                                   e90f
  while (scanf("%d%d",&n,&m)!=EOF)               6853
  {                                              4506
    memset(map,0,sizeof(map));                   0ea5
    while (m——)                                  c864
    {                                            4506
     scanf("%d%d%d",&i,&j,&k);                   0f52
     map[i][j]+=k;                               685d
     map[j][i]+=k;                               24d1
    }                                            95cf
    int mint=999999999;                          575e
    while (n>1)                                  65b7
    {                                            4506
     k=mincut();                                 2e7a
     if (k<mint) mint=k;                         4efe
     contract(sx,tx);                            d185
    }                                            95cf
    printf("%d\n",mint);                         0f28
  }                                              95cf
  return 0;                                      7021
```

```
95cf }
```

6.12 度限制生成树

```
e0a5 #include <iostream>
ef0e #include <fstream>
9581 #include <climits>
acb9 #include <queue>
8c52 #include <map>
ef2f #include <cstring>
2349 #include <string>
421c using namespace std;
d8fa const int maxn = 25;
65a1 struct node
4506 {
7a26   int v, w;
329b };
a433 struct cmp
4506 {
3d46   bool operator() (const node &a, const node &b)
4506   {
1d47     return a.w > b.w; // 是从小到大>.
95cf   }
329b };
53ee int n, m, s; // n 个点，m 条边，s 为原点. 点为有度数限制的点V0
701e int num;
4111 int minV0[maxn];
84b0 int total; // 限制的度数.
e8c8 int dist[maxn];
4363 int g[maxn][maxn]; // 用二维数组来记录图。
aede bool p[maxn];
627b int pre[maxn];
7680 int max_value[maxn], max_value_v[maxn];
172a priority_queue <node, vector<node>, cmp> Q;
bf86 map <string, int> Map;
9507 int ans;
1e17 void Prim(void);
4a0b void Solve(void);
d02f void Cal_max_value(int t);
8a96 int main(void)
4506 {
576f   int i, j;
f661   string name1, name2;
d900   int a, b, w;
769e   map <string, int>::iterator iter;
427e   // 初始化。
d8f6   Map.clear();
f041   Map["Park"] = 0;
deb8   for (i = 0; i <= maxn - 1; i++)
4506   {
f099     dist[i] = INT_MAX;
    pre[i] = -1;                                      1a39
    for (j = 0; j <= maxn - 1; j++)                   5037
    {                                                 4506
      g[i][j] = INT_MAX;                              b4df
    }                                                 95cf
  }                                                   95cf
  n = 0;                                              91c9
  cin >> m;                                           2eb3
  for (i = 1; i <= m; i++)                            6988
  {                                                   4506
   cin >> name1 >> name2 >> w;                        fad5
   iter = Map.find(name1);                            a3dd
   if (iter == Map.end())                             917d
   { // 说明该结点还不存在.                              4506
     n++;                                             c93c
     Map[name1] = n;                                  8edc
   }                                                  95cf
   a = Map[name1];                                    fbab
   iter = Map.find(name2);                            a84b
   if (iter == Map.end())                             917d
   { // 说明该结点还不存在.                              4506
     n++;                                             c93c
     Map[name2] = n;                                  e2a7
   }                                                  95cf
   b = Map[name2];                                    3878
   if (g[a][b] > w)                                   a996
   {                                                  4506
     g[a][b] = g[b][a] = w;                           ff0f
   }                                                  95cf
  }                                                   95cf
  cin >> total; // 输入限制的度数.                       ff36
   memset(p, 0, sizeof(p)); num = 0;                  783a
   for (i = 1; i <= n; i++)                           1f5c
   {                                                  4506
    if (!p[i])                                        0dfd
    {                                                 4506
      s = i;                                          ed5b
      num++; minV0[num] = s;                          303a
     // 求除去限制结点的最小生成树.                        427e
     Prim();                                          af5a
    }                                                 95cf
   }                                                  95cf
  ans = 0;                                            7360
  for (i = 1; i <= n; i++) ans += dist[i];            67a2
  // 求最小度限制生成树.                                  427e
  Solve();                                            ceca
  printf("Total miles driven: %d\n", ans);            d72b
  return 0;                                           7021
}                                                     95cf
void Prim(void)                                       103f
{                                                     4506
```

```
c8ed    int i, j, k;
3495    node mini, temp;
a058    while (!Q.empty()) Q.pop();
c7c1    dist[s] = 0;
a8ed    temp.v = s; temp.w = 0;
6b09    Q.push(temp);
e0e5    for (k = 1; k <= n; k++)
4506    {
1b18      while (!Q.empty())
4506      {
c3cc        mini = Q.top();
f2f8        Q.pop();
6e31        j = mini.v;
6dbf        if (!p[j])
4506        {
cde5          p[j] = 1;
37a1          if (g[0][j] < g[0][minV0[num]])
4506          {
0423            minV0[num] = j;
95cf          }
1f5c          for (i = 1; i <= n; i++)
4506          {
ae84            if (i != j && !p[i] && dist[i]
                  > g[j][i])
4506            {
cf01              dist[i] = g[j][i];
671b              pre[i] = j;
d413              temp.w = dist[i]; temp.v = i;
6b09              Q.push(temp);
95cf            }
95cf          }
6173          break;
95cf        }
95cf      }
95cf    }
95cf  }
427e
4abc  void Cal_max_value(int t)
4506  {
c8ed    int i, j, k;
bb7c    int Stack[maxn];
22df    int top(-1);
d46a    i = t;
437a    while (pre[i] != 0 && pre[i] != -1)
4506    {
0d63      p[i] = 1;
6a17     Stack[++top] = i;
59b2     i = pre[i];
95cf    }
9047    if (top < 0) return ;
f6a3    j = Stack[top];
abf1    max_value[j] = g[j][pre[j]];
8b10    max_value_v[j] = j;
```

```
                                                  427e
  for (i = top - 1; i >= 0; i--)                  990e
  {                                               4506
   j = Stack[i]; k = Stack[i + 1];                5cac
   if (max_value[k] > g[j][pre[j]])               74d2
   {                                              4506
    max_value[j] = max_value[k];                  30b8
    max_value_v[j] = max_value_v[k];              4b3f
   }                                              95cf
   else                                           649a
   {                                              4506
    max_value[j] = g[j][pre[j]];                  abf1
    max_value_v[j] = j;                           8b10
   }                                              95cf
  }                                               95cf
}                                                 95cf
void Solve(void)                                  a038
{                                                 4506
  int i, j, k, l;                                 80a8
  int mini, opti_i, opti_maxV;                    9059
.                                                 dc9d
  for (k = 1; k <= num; k++)                      0e77
  {                                               4506
   ans += g[0][minV0[k]];                         9bcf
                                                  427e
  j = minV0[k]; i = pre[j];                       8d9b
   while (i != -1)                                c32d
   {                                              4506
    l = i;                                        ae09
    i = pre[l];                                   00ff
    pre[l] = j;                                   034d
    j = l;                                        0ffc
   }                                              95cf
   pre[minV0[k]] = 0;                             ad40
  }                                               95cf
                                                  427e
  memset(p, 0, sizeof(p));                        ce2a
  for (i = 1; i <= n; i++)                        1f5c
  {                                               4506
   if (!p[i])                                     0dfd
   {                                              4506
    Cal_max_value(i);                             5213
   }                                              95cf
  }                                               95cf
  for (k = 1; k <= total - num; k++)              3f24
  {                                               4506
   mini = 0;                                      a722
   for (i = 1; i <= n; i++)                       1f5c
   {                                              4506
    if (pre[i] == 0) continue;                    33ca
                                                  427e
    if (g[0][i] - max_value[i] < mini)            9639
    {                                             4506
2a0e       mini = g[0][i] - max_value[i];
8b85       opti_i = i; opti_maxV = max_value_v[i];
95cf     }
95cf   }
6bbd   if (mini == 0) break;
dc8c   ans += mini;
427e
adda   pre[opti_maxV] = -1;
f116   j = opti_i; i = pre[j];
c32d   while (i != -1)
4506   {
ae09     l = i;
00ff     i = pre[l];
034d     pre[l] = j;
0ffc     j = l;
95cf   }
3647   pre[opti_i] = 0;
55fa   Cal_max_value(opti_maxV);
95cf  }
95cf }
```

6.13 最小树形图

```
e0a5 #include <iostream>
59b9 #include <cstdio>
ef2f #include <cstring>
c928 #include <cmath>
bfb3 #define MAXN 128
b080 #define MAXM 32768
2f38 #define INF 1e15
421c using namespace std;
4674 double g[MAXN][MAXN];
6849 double res;
35b8 int n, m;
935a double sqr(double x)
4506 {
ef78   return x * x;
95cf }
2d75 double dist(double xa, double ya, double xb, double yb)
4506 {
5550   return sqrt(sqr(xa - xb) + sqr(ya - yb));
95cf }
18ce void print_map()
4506 {
2ad4   for (int i = 1; i <= n; ++i)
4506   {
8c5b     for (int j = 1; j <= n; ++j)
03d4       printf("%.2f ", g[i][j]);
00e2     printf("\n");
95cf   }
95cf }
```

```
bool init()                                              28f2
{                                                        4506
  if (scanf("%d%d", &n, &m) == EOF)                      f547
    return false;
  double x[MAXN], y[MAXN];                               eec6
  int a, b;                                              e635
  for (int i = 1; i <= n; ++i)                           2ad4
    scanf("%lf%lf", &x[i], &y[i]);                       3c14
  for (int i = 1; i <= n; ++i)                           2ad4
    for (int j = 1; j <= n; ++j)                         8c5b
      g[i][j] = INF;                                     f5a0
  for (int i = 1; i <= m; ++i)                           2af5
  {                                                      4506
    scanf("%d%d", &a, &b);                               a6b8
    g[a][b] = min(g[a][b], dist(x[a], y[a], x[b], y[b])); 3126
  }                                                      95cf
  //print_map();                                         427e
  return true;                                           3361
}                                                        95cf
bool vst[MAXN];                                          c753
void dfs(int v)                                          8aca
{                                                        4506
  vst[v] = true;                                         84ad
  for (int i = 1; i <= n; ++i)                           2ad4
    if (!vst[i] && g[v][i] != INF) dfs(i);               7532
}                                                        95cf
bool possible(int v)                                     1a66
{                                                        4506
  memset(vst, false, sizeof(vst));                       ddf0
  dfs(v);                                                5f3c
  for (int i = 1; i <= n; ++i)                           2ad4
    if (i != v && !vst[i]) return false;                 48aa
  return true;                                           3361
}                                                        95cf
int pre[MAXN];                                           fe84
bool del[MAXN];                                          992d
void solve(int v)// 根为v                                2656
{                                                        4506
  res = 0;                                               f29e
  int num = n;                                           b1b7
  memset(del, false, sizeof(del));                       cec3
  while(1)                                               1f75
  {                                                      4506
    int i;                                               a0f7
    // 更新数组pre                                       427e
    for (i = 1; i <= n; ++i)                             0428
    {                                                    4506
      if (del[i] || i == v) continue;                    50c1
      pre[i] = i;                                        86f4
      g[i][i] = INF;                                     f900
      for (int j = 1; j <= n; ++j)                       8c5b
4506      {
fff9        if (del[j]) continue;
1f4f        if (g[j][i] < g[pre[i]][i])
671b          pre[i] = j;
95cf      }
95cf    }
0428    for (i = 1; i <= n; ++i)
4506    {
427e      // 找环
50c1      if (del[i] || i == v) continue;
8541      int j = i;
fa64      memset(vst, 0, sizeof(vst));
ee9d      while (!vst[j] && j != v)
4506      {
99c6        vst[j] = true;
b66d        j = pre[j];
95cf      }
24ac      if (j == v) continue;
934c      i = j;
427e      // 更新，有向环缩点res
b0c8      res += g[pre[i]][i];
348f      for(j = pre[i]; j != i; j = pre[j])
4506      {
6ac1        res += g[pre[j]][j];
2280        del[j] = true;
95cf      }
4b26      for(j = 1; j <= n; ++j)
4506      {
fff9        if(del[j]) continue;
a1f5        if(g[j][i] != INF)
48d3          g[j][i] –= g[pre[i]][i];
95cf      }更新缩点以后的有向环和其他点的边权
427e      //
348f      for(j = pre[i]; j != i; j = pre[j])
4506      {
ed11        for(int k = 1; k <= n; ++k)
4506        {
c844          if(del[k])continue;
170a          g[i][k] = min(g[i][k], g[j][k])
                ;
fc80          if(g[k][j] != INF)
664f            g[k][i] = min(g[k][i], g[k][j
                  ] – g[pre[j]][j]);
95cf        }
95cf      }
427e      // 完成缩点
348f      for(j = pre[i]; j != i; j = pre[j])
4506      {
2280        del[j] = true;
95cf      }
6173      break;
95cf    }
427e    // 不存在有向环时，停止循环，得出最终值res
    if(i > n){                                 6bff
      for(int i = 1; i <= n; ++i)              2ad4
      {                                        4506
        if(del[i] || i == v) continue;         50c1
        res += g[pre[i]][i];                   b0c8
      }                                        95cf
      break;                                   6173
    }                                          95cf
  }                                            95cf
}                                              95cf
int main()                                     299c
{                                              4506
  while (init())                               1f56
  {                                            4506
    if (!possible(1)) printf("poor snoopy      a248
      \n");
    else                                       649a
    {                                          4506
      solve(1);                                1d60
      printf("%.2f\n", res);                   d6bf
    }                                          95cf
  }                                            95cf
  return 0;                                    7021
}                                              95cf
```

6.14 多重匹配

```
#include <stdio.h>                             1915
const int maxn=100;                            0cbb
int a[maxn],b[maxn],nov_a[maxn],nov_b[         a23c
  maxn];
int w[maxn][maxn];                             bac0
int m,n;                                       4d9b
bool find(int i)                               6271
{                                              4506
  nov_a[i]=0;                                  a274
  for(int p=0;p<n;p++)                         b851
  if(w[i][p]==0&&nov_b[p])                     854a
  {                                            4506
    nov_b[p]=0;                                5a4d
    if(b[p]>0)                                 226a
    {                                          4506
      b[p]––;                                  aefb
      w[i][p]=1;                               060a
      return true;                             3361
    }                                          95cf
    for(int q=0;q<m;q++)                       2bc2
    if(w[q][p]==1&&nov_a[q])                   1ad5
    {                                          4506
      if(find(q))                              b5be
      {                                        4506
        w[i][p]=1;                             060a
        w[q][p]=0;                             0bb0
3361          return true;
95cf        }
95cf      }
95cf    }
438e    return false;
95cf  }
653c  bool gao(int i)
4506  {
e4c8    for(int j=0;j<m;j++)nov_a[j]=1;
66c3    for(int j=0;j<n;j++)nov_b[j]=1;
9044    return find(i);
95cf  }
299c  int main()
4506  {
e287    int cas=0;
a41b    while(scanf("%d%d",&m,&n)!=EOF&&(m||n))
4506    {
25fe      for(int i=0;i<m;i++)scanf("%d",&a[i])
            ;
e951      for(int i=0;i<n;i++)scanf("%d",&b[i])
            ;
548e      for(int i=0;i<m;i++)
6bf3      for(int j=0;j<n;j++)
dc5e      w[i][j]=0;
6e7e      bool ans=true;
548e      for(int i=0;i<m;i++)
4506      {
9212        while(a[i]>0&&gao(i))a[i]——;
4fe1        if(a[i]!=0)
4506        {
539e          ans=false;
6173          break;
95cf        }
95cf      }
f5a9      for(int i=0;i<n;i++)
e0ad      if(b[i]>0)ans=false;
548e      for(int i=0;i<m;i++)
6bf3      for(int j=0;j<n;j++)
4b54      if(w[i][j]==1)
4506      {
76a8        b[j]++;
bef5        w[i][j]=—1;
73cd        if(!gao(i))
4506        {
292e          w[i][j]=1;
cd7b          b[j]=0;
95cf        }
427e
95cf      }
6c4d      else w[i][j]=—1;
a27f      cas++;
a3dd      if(cas!=1)printf("\n");
a8ea      if(ans)
```

```
    {                                         4506
      for(int i=0;i<m;i++)                    548e
      {                                       4506
        for(int j=0;j<n;j++)                  6bf3
        if(w[i][j]==1)printf("Y");            a417
        else printf("N");                     8f87
        printf("\n");                         00e2
      }                                       95cf
    }                                         95cf
    else printf("Impossible\n");              edea
  }                                           95cf
}                                             95cf
```

7 java 样例

7.1 java 样例

```
import java.io.*;                             84fe
import java.math.BigInteger;                  93c2
import java.util.*;                           4156
public class Main {                           788a
  public static void main(String[] args)      e1b6
    {
    Scanner cin = new Scanner(new             f75d
      BufferedInputStream(System.in));
    while (cin.hasNext()){                    ac68
      int m = cin.nextInt();                  93f1
      int n = cin.nextInt();                  4f78
      int best = 0;                           f3f7
      BigInteger b[] = new BigInteger[m];     3c41
      for (int i = 0; i < m; ++i)             e725
        b[i] = BigInteger.valueOf(1);         734a
      for (int i = 0; i < n; ++i){            6c2f
        //BigInteger c = BigInteger.          427e
          valueOf(1);
        for (int j = 0; j < m; ++j){          6613
          int x = cin.nextInt();              4677
          BigInteger d = BigInteger.          b9b5
            valueOf(x);
          b[j] = b[j].multiply(d);            0541
        }                                     95cf
      }                                       95cf
      for (int i = 1; i < m; ++i){            bf14
        if (b[best].compareTo(b[i]) <= 0)     ecee
           best = i;
      }                                       95cf
      System.out.println(best + 1);           004f
                                              427e
    }                                         95cf
  }                                           95cf
                                              427e
}                                             95cf
```

8 其他

8.1 校赛 Meeting 标程

```
1915 #include <stdio.h>
1fa4 #include <math.h>
427e
842a int u1, v1, u2, v2, u3, v3;
8f9d double x, y, d;
427e
4d45 double f(double u, double v)
4506 {
ec32    double r=0;
e2fe    r+=sqrt((u-u1)*(u-u1)+(v-v1)*(v-v1));
3df7    r+=sqrt((u-u2)*(u-u2)+(v-v2)*(v-v2));
b00a    r+=sqrt((u-u3)*(u-u3)+(v-v3)*(v-v3));
05ee    if(r<d)
4506    {
5c82       d=r;
c0ce       x=u;
583c       y=v;
95cf    }
547e    return r;
95cf }
427e
afdd double f(double u)
4506 {
918b    double L, R, M1, M2;
b4a4    for(L=-1001, R=1001; R-L>1e-6; )
```

```
   {                                                4506
      M1=(L*2+R)/3;                                 3dd0
      M2=(L+R*2)/3;                                 8a02
      f(u, M1)<f(u, M2)?(R=M2):(L=M1);              40c0
   }                                                95cf
   return f(u, (L+R)/2);                            39f8
}                                                   95cf
                                                    427e
int main()                                          299c
{                                                   4506
   int T;                                           9523
   double L, R, M1, M2;                             918b
                                                    427e
   for(scanf("%d", &T); T--; )                      ba03
   {                                                4506
      scanf("%d %d %d %d %d %d", &u1, &v1           e322
        , &u2, &v2, &u3, &v3);
      for(d=1e100, L=-1001, R=1001; R-L>1           6f5e
        e-6; )
      {                                             4506
         M1=(L*2+R)/3;                              3dd0
         M2=(L+R*2)/3;                              8a02
         f(M1)<f(M2)?(R=M2):(L=M1);                 7091
      }                                             95cf
      printf("(%.3lf,%.3lf)\n", x, y);              a0d8
   }                                                95cf
                                                    427e
   return 0;                                        7021
}                                                   95cf
```