

ACM/ICPC 代码模板库

南京大学 ACM/ICPC 集训队

目录

1 比赛配置	2	4.7 MillerRabin	28
1.1 代码库校验和	2	4.8 PollardRho	28
1.2 vimrc	2	4.9 矩阵基础	29
1.3 外挂	2	4.10 高斯消元	30
2 二维计算几何	2	5 数据结构	30
2.1 定义	2	5.1 SplayTree	30
2.2 定义	3	5.2 SplayTree	35
2.3 点与线	4	5.3 kdtree	38
2.4 三角形	8	5.4 后缀数组	40
2.5 多边形	9	5.5 线段树区间修改	41
2.6 面积	13	5.6 AhoCorasick	42
2.7 球面	14	5.7 kmp	43
2.8 圆	14	5.8 Palindromic	44
2.9 网格	17	6 图论	44
2.10 区域中点集个数	17	6.1 Hamilton	44
3 三维计算几何	19	6.2 HopcraftKarp	45
3.1 定义	19	6.3 HopcraftKarp	46
3.2 点线面	20	6.4 Hungary	47
3.3 面积	24	6.5 KM	47
3.4 体积	25	6.6 lca	48
3.5 重心	25	6.7 MaxcostMaxflow	49
3.6 凸包	25	6.8 NetworkFlow	50
4 数论	26	6.9 SCC	51
4.1 头文件	26	6.10 Vconnect	52
4.2 基础	26	6.11 Stoer-Wagne	55
4.3 线性筛法	26	6.12 度限制生成树	55
4.4 线性同余方程	27	6.13 最小树形图	57
4.5 中国剩余定理	27	6.14 多重匹配	59
4.6 离散对数	27	7 java 样例	60
		7.1 java 样例	60

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\\s\\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 外挂

```
427e // 调栈空间
08e0 const int N_MAX = 10000000;
772b static int stack[N_MAX * 5], bak;
68df asm __volatile__
bbf4 (
4d52     "movl %%esp, %0;"
```

```
"movl %1, %%esp;":
      22d3
"=g"(bak):
      7f26
"=g"(stack + N_MAX * 5 - 1):
      5bf1
);
      fe62
427e
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++; \
7181     while (((*pt) >= '0' && (*pt) <= '9'))
      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

```
#define eps 1e-8
652e #define fabs(x) ((x) > 0? (x): -(x))
c1b0 #define zero(x) (fabs(x) < eps)
0102 #define _sign(x) ((x)>eps?1:((x)<-eps
12d8 ?2:0))
dca2 #define sqr(x) ((x)*(x))
418f #define MAXN 1000
8d0a #define offset 10000
13f1 const double pi=acos(-1);
427e
427e // 点的定义
427e struct point{
9704     int index;
082e     double ang;
98c9     double x, y;
d0aa     point(){x = 0;y = 0;}
52a2     point(double sx, double sy){
df98         x = sx;
e87b         y = sy;
d22b     }
95cf     void read(){
4f13
```

```

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

struct line{
    double ang;
    point a, b;line(){};
    line(const point &p1, const point &p2){
        a = p1;
        b = p2;
    }
    bool operator < (const line &y)const{
        if (zero(ang - y.ang))
            return (xmult(a, y.b, y.a) < 0);
        return ang < y.ang;
    }
    void getang(){
        ang = atan2(b.y - a.y, b.x - a.x);
    }
    friend ostream& operator << (ostream &
        out, const line &a);
};
ostream& operator << (ostream &out, const
    line &a){
    out<<a.a<<' '<<a.b<<' '<<a.ang;
    return out;
}
typedef const line CL;
// 圆定义
struct circle{
    double r;
    point c;
    circle(){};
    circle(const point &p, double x){
        c = p;
        x = r;
    }
    friend ostream& operator << (ostream &
        out, const circle &a);
};
ostream& operator << (ostream &out, const
    circle &a){
    out<<a.c<<' '<<a.r;
    return out;
}
typedef const circle CC;

2.2 定义

#define eps 1e-8
#define fabs(x) ((x) > 0? (x): -(x))
#define zero(x) (fabs(x) < eps)
#define _sign(x) ((x)>eps?1:((x)<-eps
    ?2:0))
#define MAXN 1000
#define offset 10000
#define sqr(x) ((x)*(x))
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;
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 };
05c6 ostream& operator << (ostream &out, const
df9c     point &a){
        out<<a.x<<' '<<a.y;

        return out;
    }
    bool cmp(const point &p1, const point &p2)
    ){
        return p1.ang < p2.ang;
    }

    // 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
    struct line{
        double ang;
        point a, b;line(){};
        line(const point &p1, const point &p2){
            a = p1;
            b = p2;
        }
        bool operator < (const line &y)const{
            if (zero(ang - y.ang))
                return (xmult(a, y.b, y.a) < 0);
            return ang < y.ang;
        }
        void getang(){
            ang = atan2(b.y - a.y, b.x - a.x);
        }
        friend ostream& operator << (ostream &
            out, const line &a);
    };
    ostream& operator << (ostream &out, const
        line &a){
        out<<a.a<<' '<<a.b<<' '<<a.ang;
        return out;
    }

    // 圆定义
    struct circle{
        double r;
        point c;
        circle(){};
        circle(const point &p, double x){
            c = p;
            x = r;
        }
        friend ostream& operator << (ostream &
            out, const circle &a);
    };
    ostream& operator << (ostream &out, const
        circle &a){
        out<<a.c<<' '<<a.r;
        return out;
    }
}

2.3 点与线

// 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){
7c36     return sqrt((v.x * v.x) + (v.y * v.y));
95cf }
427e
427e // 两点距离
21ca double dis(CP &p1,CP &p2){
b08a     return sqrt((p1.x-p2.x)*(p1.x-p2.x)+(p1
        .y-p2.y)*(p1.y-p2.y));
95cf }
8046 double dis(double x1,double y1,double x2,
        double y2){
9f44     return sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1
        -y2));
95cf }
1627 double dis2(CP &p1,CP &p2){
3eaf     return (p1.x-p2.x)*(p1.x-p2.x)+(p1.y-p2
        .y)*(p1.y-p2.y);
95cf }
a880 double dis2(double x1,double y1,double x2
        ,double y2){
5819     return (x1-x2)*(x1-x2)+(y1-y2)*(y1-y2);
95cf }
427e
427e // 判三点共线
68d7 int dots_inline(CP &p1,CP &p2,CP &p3){
20b6     return zero(xmult(p1,p2,p3));
95cf }

int dots_inline(double x1,double y1,
double x2,double y2,double x3,double y3
){
    return zero(xmult(x1,y1,x2,y2,x3,y3));
}
// 判点是否在线段上, 包括端点
int dot_online_in(CP &p,CL &l){
    return zero(xmult(p,l.a,l.b))&&(l.a.x-p
.x)*(l.b.x-p.x)<eps&&(l.a.y-p.y)*(l.b
.y-p.y)<eps;
}
int dot_online_in(CP &p,CP &l1,CP &l2){
    return zero(xmult(p,l1,l2))&&(l1.x-p.x)
*(l2.x-p.x)<eps&&(l1.y-p.y)*(l2.y-p.y
)<eps;
}
int dot_online_in(double x,double y,
double x1,double y1,double x2,double y2
){
    return zero(xmult(x,y,x1,y1,x2,y2))&&(
x1-x)*(x2-x)<eps&&(y1-y)*(y2-y)<eps;
}
// 判点是否在线段上不包括端点,
int dot_online_ex(CP &p,CL &l){
    return dot_online_in(p,l)&&(!zero(p.x-l
.a.x)||!zero(p.y-l.a.y))&&(!zero(p.x-
l.b.x)||!zero(p.y-l.b.y));
}
int dot_online_ex(CP &p,CP &l1,CP &l2){
    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));
}
int dot_online_ex(double x,double y,
double x1,double y1,double x2,double y2
){
    return dot_online_in(x,y,x1,y1,x2,y2)
&&(!zero(x-x1)||!zero(y-y1))&&(!zero(
x-x2)||!zero(y-y2));
}
// 判两点在线段同侧, 点在线段上返回0
int same_side(CP &p1,CP &p2,CL &l){
    return xmult(l.a,p1,l.b)*xmult(l.a,p2,l
.b)>eps;
}
int same_side(CP &p1,CP &p2,CP &l1,CP &l2
){
    return xmult(l1,p1,l2)*xmult(l1,p2,l2)>
eps;
}

```

```

427e // 判两点在线段异侧, 点在线段上返回0
427e int opposite_side(CP &p1, CP &p2, CL &l){
b5f2     return xmult(l.a, p1, l.b)*xmult(l.a, p2, l
95bc         .b)<-eps;
95cf }
a050 int opposite_side(CP &p1, CP &p2, CP &l1, CP
      &l2){
de06     return xmult(l1, p1, l2)*xmult(l1, p2, l2)
      <-eps;
95cf }
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 // 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 }
518c int perpendicular(CP &u1, CP &u2, CP &v1, CP
      &v2){
6143     return zero((u1.x-u2.x)*(v1.x-v2.x)+(u1
      .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))
d666         return !same_side(u.a, u.b, v)&&!
          same_side(v.a, v.b, u);
efeb     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);
95cf }
2882 int intersect_in(CP &u1, CP &u2, CP &v1, CP
      &v2){
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);
      }
      // 判两线段相交, 不包括端点和部分重合
      int intersect_ex(CL &u, CL &v){
          return opposite_side(u.a, u.b, v)&&
              opposite_side(v.a, v.b, u);
      }
      int intersect_ex(CP &u1, CP &u2, CP &v1, CP
          &v2){
          return opposite_side(u1, u2, v1, v2)&&
              opposite_side(v1, v2, u1, u2);
      }
      // 计算两直线交点, 注意事先判断直线是否平行!
      // 线段交点请另外判线段相交, 同时还是要判断
      是否平行
      point intersection(CL &u, CL &v){
          point ret=u.a;
          double t=((u.a.x-v.a.x)*(v.a.y-v.b.y)-(
              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
                  .y-u.b.y)*(v.a.x-v.b.x));
          ret.x+=(u.b.x-u.a.x)*t;
          ret.y+=(u.b.y-u.a.y)*t;
          return ret;
      }
      point intersection(CP &u1, CP &u2, CP &v1,
          CP &v2){
          point ret=u1;
          double t=((u1.x-v1.x)*(v1.y-v2.y)-(u1.y
              -v1.y)*(v1.x-v2.x))
              /((u1.x-u2.x)*(v1.y-v2.y)-(u1.y-u2
                  .y)*(v1.x-v2.x));
          ret.x+=(u2.x-u1.x)*t;
          ret.y+=(u2.y-u1.y)*t;
          return ret;
      }
      // 求给定线段的中垂线
      line pbline(CL &l){
          line ret; ret.a = (l.a + l.b) / 2;
          double a = l.b.x - l.a.x, b = l.b.y - l
              .a.y;
          double c = (l.a.y - l.b.y) * ret.a.y +
              (l.a.x - l.b.x) * ret.a.x;
          if (!zero(a)){
              ret.b.y = 0; ret.b.x = -c / a;
              if (zero(dis(ret.a, ret.b))){
                  ret.b.y = 1e10; ret.b.x = -(c - b *
                      ret.b.y) / a;
              }
          }
          else{
              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){
        point t=p;
        t.x+=l.a.y-l.b.y,t.y+=l.b.x-l.a.x;
        if (xmult(l.a,t,p)*xmult(l.b,t,p)>eps)
            return dis(p,l.a)<dis(p,l.b)?dis(p,l
                a):dis(p,l.b);
        return fabs(xmult(p,l.a,l.b))/dis(l.a,l
            .b);
    }
    double disptoseg(CP &p,CP &l1,CP &l2){
        point t=p;
        t.x+=l1.y-l2.y,t.y+=l2.x-l1.x;
        if (xmult(l1,t,p)*xmult(l2,t,p)>eps)
            return dis(p,l1)<dis(p,l2)?dis(p,l1):
                dis(p,l2);
        return fabs(xmult(p,l1,l2))/dis(l1,l2);
    }
    // 线段到线段距离, 事先判断相交情况
    double dissegtoseg(CL &l1, CL &l2){
        return min(min(disptoseg(l1.a, l2),
            disptoseg(l1.b, l2)), min(disptoseg(
                l2.a, l1), disptoseg(l2.b, l1)));
    }
    double dissegtoseg(CP &l1a, CP &l1b, CP &
        l2a, CP &l2b){
        return min(min(disptoseg(l1a, l2a, l2b)
            , disptoseg(l1b, l2a, l2b)), min(
                disptoseg(l2a, l1a, l1b), disptoseg(
                    l2b, l1a, l1b)));
    }
    // 矢量V 以P 为顶点逆时针旋转angle 并放
    大scale 倍
    point rotate(point v,point p,double angle
        ,double scale){
        point ret=p;
        v.x=-p.x,v.y=-p.y;
        p.x=scale*cos(angle);
        p.y=scale*sin(angle);
        ret.x+=v.x*p.x-v.y*p.y;
        ret.y+=v.x*p.y+v.y*p.x;
        return ret;
    }
    // p 在新坐标系O(I,e1,e2) 中的坐标
    point rotate(CP &p, CP &I, CP &e1, CP &e2
        ){
        point p2;
        p2.x = I.x + e1.x * p.x + e1.y * p.y;
        p2.y = I.y + e2.x * p.x + e2.y * p.y;
        return p2;
    }

```

```

427e //p 点绕原点按逆时针旋转angle
427e point rotate(CP &p, double angle){
3f0e     point e1, e2, I;
3e07     e1.x = cos(angle);e1.y = -sin(angle);
1f1c     e2.x = -e1.y;e2.y = e1.x;
ed21     I.x = 0;I.y = 0;
085d     return rotate(p, I, e1, e2);
e586 }
95cf }
427e // 返回值[0, 4), 正比向量v1 到向量v2 的顺时
427e // 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

```

427e // 外心
878f point circumcenter(CP &a,CP &b,CP &c){
7173     line u,v;
2ecf     u.a.x=(a.x+b.x)/2;
5a23     u.a.y=(a.y+b.y)/2;
ee31     u.b.x=u.a.x-a.y+b.y;
77b1     u.b.y=u.a.y+a.x-b.x;
5396     v.a.x=(a.x+c.x)/2;
8492     v.a.y=(a.y+c.y)/2;
abe9     v.b.x=v.a.x-a.y+c.y;
63ff     v.b.y=v.a.y+a.x-c.x;
ed5d     return intersection(u,v);
95cf }
427e // 内心
6bd8 point incenter(CP &a,CP &b,CP &c){
7173     line u,v;
1a7c     double m,n;
2aec     u.a=a;
cd0a     m=atan2(b.y-a.y,b.x-a.x);
4211     n=atan2(c.y-a.y,c.x-a.x);
af8c     u.b.x=u.a.x+cos((m+n)/2);

```

```

u.b.y=u.a.y+sin((m+n)/2);
v.a=b;
m=atan2(a.y-b.y,a.x-b.x);
n=atan2(c.y-b.y,c.x-b.x);
v.b.x=v.a.x+cos((m+n)/2);
v.b.y=v.a.y+sin((m+n)/2);
return intersection(u,v);
}
// 垂心
point perpendcenter(CP &a,CP &b,CP &c){
line u,v;
u.a=c;
u.b.x=u.a.x-a.y+b.y;
u.b.y=u.a.y+a.x-b.x;
v.a=b;
v.b.x=v.a.x-a.y+c.y;
v.b.y=v.a.y+a.x-c.x;
return intersection(u,v);
}
// 重心
// 到三角形三顶点距离的平方和最小的点
// 三角形内到三边距离之积最大的点
point barycenter(CP &a,CP &b,CP &c){
line u,v;
u.a.x=(a.x+b.x)/2;
u.a.y=(a.y+b.y)/2;
u.b=c;
v.a.x=(a.x+c.x)/2;
v.a.y=(a.y+c.y)/2;
v.b=b;
return intersection(u,v);
}
// 费马点
// 到三角形三顶点距离之和最小的点
point fermentpoint(CP &a,CP &b,CP &c){
point u,v;
double step=fabs(a.x)+fabs(a.y)+fabs(b.
x)+fabs(b.y)+fabs(c.x)+fabs(c.y);
int i,j,k;
u.x=(a.x+b.x+c.x)/3;
u.y=(a.y+b.y+c.y)/3;
while (step>1e-10)
for (k=0;k<10;step/=2,k++)
for (i=-1;i<=1;i++)
for (j=-1;j<=1;j++){
v.x=u.x+step*i;
v.y=u.y+step*j;
if (dis(u,a)+dis(u,b)+dis(u,c)>
dis(v,a)+dis(v,b)+dis(v,c))
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 // 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 // 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
ea98 int inside_convex(CP &q, int n, point* p){
53c4     int i, s[3]={1,1,1};
cb40     for (i=0; i<n&&s[1]|s[2]; i++)
8fa5         s[_sign(xmult(p[(i+1)%n], q, p[i]))]=0;
e8ad     return s[1]|s[2];
95cf }

427e // 判点在凸多边形内, 顶点按顺时针或逆时针给
      出, 在多边形边上返回0
9e0e int inside_convex_v2(CP &q, 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++)
8fa5         s[_sign(xmult(p[(i+1)%n], q, p[i]))]=0;
e92f     return s[0]&&s[1]|s[2];
95cf }

427e // 判点在任意多边形内顶点按顺时针或逆时针给
      出,
427e // on_edge 表示点在多边形边上时的返回
      值, offset 为多边形坐标上限
78bd int inside_polygon(CP &q, int n, point* p,
    int on_edge=1){
af1a     point q2;
1adf     int i=0, count;
22c1     while (i<n)

```

```

      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
              [(i+1)%n], q2)<-eps&&xmult(p[i], q,
              p[(i+1)%n])*xmult(p[i], q2, p[(i+1)
              %n])<-eps)  f875
              count++;
          return count&1; 45d2
      } 9103

// 判线段在任意多边形内, 顶点按顺时针或逆时
// 针给出, 与边界相交返回1 95cf
int inside_polygon(CP &l1, CP &l2, int n,  427e
    point* p){ 427e
    point t[MAXN], tt;
    int i, j, k=0;
    if (!inside_polygon(l1, n, p)||!
        inside_polygon(l2, n, p))
        return 0; 7021
    for (i=0; i<n; i++) 2dbf
        if (opposite_side(l1, l2, p[i], p[(i+1)%
            n])&&opposite_side(p[i], p[(i+1)%n],
            l1, l2)) 6a8f
            return 0; 7021
        else if (dot_online_in(l1, p[i], p[(i
            +1)%n])) d6e2
            t[k++]=l1; 29be
        else if (dot_online_in(l2, p[i], p[(i
            +1)%n])) cbea
            t[k++]=l2; 3d75
        else if (dot_online_in(p[i], l1, l2)) 1abb
            t[k++]=p[i]; 8b2e
    for (i=0; i<k; i++) a83a
        for (j=i+1; j<k; j++){ e784
            tt.x=(t[i].x+t[j].x)/2; 2874
            tt.y=(t[i].y+t[j].y)/2; abed
            if (!inside_polygon(tt, n, p)) 4b25
                return 0; 7021
        } 95cf
    return 1; 7459
} 95cf

// 多边形重心 427e
point barycenter(int n, point* p){ 427e
    point ret, t; 3a12
    double t1=0, t2; 54ec
    3444

```

```

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

```

```

4206     return ans;
95cf }
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=10000000000;
9c16     for(int p=0;p<n;++p)
4506     {
f1dc         while(xmult(con[(p+1)%n],con[(q+1)%n]
          ],con[p])<xmilt(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 = 10000000000;
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 // 求两个不包含的凸包的最短距离, 逆时针输入
427e     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 = xmilt(b[(p2 + 1) % m], a[
          p1], a[(p1 + 1) % n]);
474d         t -= xmilt(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){
          ans = min(ans, disptoseg(b[p2], a[
            p1], a[(p1 + 1) % n]));
          p1 = (p1 + 1) % n;
          }else{
            ans = min(ans, dissegtoseg(a[p1], a
              [(p1 + 1) % n], b[p2], b[(p2 + 1)
                % m]));
            p1 = (p1 + 1) % n;
            p2 = (p2 + 1) % m;
          }
        }
        return ans;
    }
    // 求多边形中最长的线段, 的长度, 线段储存
    在 l 中
double inside_polygon_max(int n,point* p,
      line &l){
double len = 0;
for (int i = 0; i < n; ++i)
    for (int j = i + 1; j < n; ++j){
        vector<point> points;
        points.clear();
        points.push_back(p[i]);
        points.push_back(p[j]);
        for (int a = 0; a < n; ++a)
            for (int b = a + 1; b < n; ++b){
                if (a == i) continue;
                if (parallel(p[i], p[j], p[a],
                    p[b]))
                    continue;
                point p1 = intersection(p[i], p
                    [j], p[a], p[b]);
                if (dmult(p[a], p[b], p1) <= 0)
                {
                    points.push_back(p1);
                }
            }
        sort(points.begin(), points.end());
        int s = 0;
        for (int k = 0; k < points.size() -
            1; ++k){
            if (zero(dis(points[k], points[k
                + 1]))) continue;
            point p1;
            p1 = (points[k] + points[k + 1])
                / 2;
            if (inside_polygon(p1, n, p))
                continue;
            double d = dis(points[s], points[
                k + 1]);
            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)
4d87         lines[i].getang();
efa0     sort(lines, lines + n);
dcba     int m = 1;
bc29     for (int i = 1; i < n; ++i)
fa79         if (!zero(lines[i].ang - lines[i -
            1].ang))
aa88             lines[m++] = lines[i];
7852     n = m;
dad4     int bot = 0, top = 1;
a007     for (int i = 2; i < n; ++i){
f9fe         if ((parallel(lines[top], lines[top -
            1]) || parallel(lines[bot], lines[
                bot + 1])))
7021             return 0;
ba77         while ((bot < top) && (xmult(
            intersection(lines[top], lines[top
                - 1]), lines[i].b, lines[i].a) >
            eps))
f959             —top;
c049         while ((bot < top) && (xmult(
            intersection(lines[bot], lines[bot
                + 1]), lines[i].b, lines[i].a) >
            eps))
5425             ++bot;
c56b         ++top;
973c         lines[top] = lines[i];
        }
        while ((bot < top) && (xmult(intersection
            (lines[top], lines[top - 1]), lines[
                bot].b, lines[bot].a) > eps))
            —top;
        while ((bot < top) && (xmult(intersection
            (lines[bot], lines[bot + 1]), lines[
                top].b, lines[top].a) > eps))
            ++bot;
        if (top <= bot + 1) return 0;
        n = 0;
        for (int i = bot; i < top; ++i)
            p[n++] = intersection(lines[i], lines
                [i + 1]);
        if (bot < top + 1)
            p[n++] = intersection(lines[bot],
                lines[top]);
        return n;
    }
    // 算法顺时针构造包含所有共线点的凸
    包graham, O(nlogn)
    point p1, p2;
    int graham_cp(const void* a, const void* b
    ){
        double ret = xmult(*((point*)a), *((point
            *)b), p1);
        return zero(ret) ? (xmult(*((point*)a)
            , *((point*)b), p2) > 0 ? 1 : -1) : (ret
            > 0 ? 1 : -1);
    }
    void _graham(int n, point* p, int& s, point*
        ch){
        int i, k = 0;
        for (p1 = p2 = p[0], i = 1; i < n; p2.x += p[i].x, p2
            .y += p[i].y, i++)
            if (p1.y - p[i].y > eps || (zero(p1.y - p[i].
                y) && p1.x > p[i].x))
                p1 = p[k = i];
        p2.x /= n, p2.y /= n;
        p[k] = p[0], p[0] = p1;
        // cout << n << endl;
        qsort(p + 1, n - 1, sizeof(point), graham_cp);
        // cout << n << endl;
        for (ch[0] = p[0], ch[1] = p[1], ch[2] = p[2], s
            = i = 3; i < n; ch[s++] = p[i++])
            for (; s > 2 && xmult(ch[s - 2], p[i], ch[s
                - 1]) < -eps; s —);
    }
    // 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!)
    // 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;
6bd4     v2.b.x = v.b.y - v.a.y + p.x;
99c4     v2.b.y = v.a.x - v.b.x + p.y;
efc1     if (dmult(v2.b - v2.a, u.b - u.a) < 0)
          swap(v2.a, v2.b);
10cd     double alpha = xmult(v2.b - v2.a, u.b -
          u.a) / len(v2.b - v2.a) / len(u.b -
          u.a);
d9eb     alpha = asin(alpha / r) + atan2(v2.b.y
          - v2.a.y, v2.b.x - v2.a.x);
57f9     v2.a = p;
2a97     v2.b.x = 10 * cos(alpha) + v2.a.x;
33d7     v2.b.y = 10 * sin(alpha) + v2.a.y;
9d48     return v2;
95cf }求入射边
427e
427e //关于凸包折射两次的情况, 出射角保存在中, 折
      射率为vvr如果不相交, 返回
427e //, false 不考虑镜面反射和入射到凸包角上的情
      况
0c20 bool refraction(int n, point p[], line &
      v, double r){
4e58     int index = -1;
c9aa     line l[n];
8ac5     point p1;
6c2f     for (int i = 0; i < n; ++i){
bd2f         l[i].a = p[i];

```

```

      l[i].b = p[(i + 1) % n];
      }
      for (int i = 0; i < n; ++i){
          if (parallel(v, l[i])) continue;
          point p2 = intersection(v, l[i]);
          if (dmult(l[i].a, l[i].b, p2) >= 0)
              continue;
          if (dmult(p2, v.b, v.a) <= 0)
              continue;
          if ((index == -1)||((dis2(p2, v.a) <
              dis2(p1, v.a)))){
              index = i;
              p1 = p2;
          }
      }
      if (index == -1) return 0;
      swap(l[0], l[index]);
      v = refraction(v, l[0], p1, r);
      index = -1;
      for (int i = 1; i < n; ++i){
          if (parallel(v, l[i])) continue;
          point p2 = intersection(v, l[i]);
          if (dmult(l[i].a, l[i].b, p2) >= 0)
              continue;
          if (dmult(p2, v.b, v.a) <= 0)
              continue;
          if ((index == -1)||((dis2(p2, v.a) <
              dis2(p1, v.a)))){
              index = i;
              p1 = p2;
          }
      }
      swap(l[1], l[index]);
      v = refraction(v, l[1], p1, 1 / r);
      return true;
  }
}

```

2.6 面积

```

// 计算三角形面积, 输入三顶点
double area_triangle(CP &p1, CP &p2, CP &p3
427e
ab5f ){
      return fabs(xmult(p1,p2,p3))/2;
      f8a3
      95cf
      }
double area_triangle(double x1,double y1,
243e double x2,double y2,double x3,double y3
      ){
      return fabs(xmult(x1,y1,x2,y2,x3,y3))
      1953
      /2;
      95cf
      }
// 计算三角形面积, 输入三边长
double area_triangle(double a,double b,
427e
427e double c){
      7fb2

```

```

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;
d991        if (dlng>pi)
ea50            dlng=pi+pi-dlng;
6a31        lat1*=pi/180,lat2*=pi/180;
dc31        return acos(cos(lat1)*cos(lat2)*cos(
            dlng)+sin(lat1)*sin(lat2));
95cf   }
427e
427e    // 计算距离,r 为球半径
3a23    double line_dist(double r,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;
d991        if (dlng>pi)
ea50            dlng=pi+pi-dlng;
6a31        lat1*=pi/180,lat2*=pi/180;
24df        return r*sqrt(2-2*(cos(lat1)*cos(lat2)*
            cos(dlng)+sin(lat1)*sin(lat2)));
95cf   }
427e
427e    // 计算球面距离,r 为球半径
1bec    inline double sphere_dist(double r,double
    lng1,double lat1,double lng2,double
    lat2){
5db0        return r*angle(lng1,lat1,lng2,lat2);
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
641a    int intersect_line_circle(CC &c,CP &l1,CP
    &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    }
e3cf    int intersect_seg_circle(CC &c,CP &l1,CP
    &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){
        return dis(c1,c2)<r1+r2+eps&&dis(c1,c2)
9676        >fabs(r1-r2)-eps;
95cf    }
da9a    int intersect_circle_circle(CC &c1,CC &c2
    ){
        return dis(c1.c,c2.c)<c1.r+c2.r+eps&&
14e2        dis(c1.c,c2.c)>fabs(c1.r-c2.r)-eps;
95cf    }
427e    // 计算圆上到点p 最近点, 如p 与圆心重合, 返
427e    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
c8d1 void intersection_line_circle(CP &c,
        double r,CP &l1,CP &l2,point& p1,point&
        p2){
e36e     point p=c;
3337     double t;
e339     p.x+=l1.y-l2.y;
4399     p.y+=l2.x-l1.x;
1b68     p=intersection(p,c,l1,l2);
d753     t=sqrt(r*r-dis(p,c)*dis(p,c))/dis(l1,l2
        );
f468     p1.x=p.x+(l2.x-l1.x)*t;
7618     p1.y=p.y+(l2.y-l1.y)*t;
d8a7     p2.x=p.x-(l2.x-l1.x)*t;
24fc     p2.y=p.y-(l2.y-l1.y)*t;
95cf }
c26e void intersection_line_circle(CC &c,CP &
        l1,CP &l2,point& p1,point& p2){
92cd     point p=c.c;
3337     double t;
e339     p.x+=l1.y-l2.y;
4399     p.y+=l2.x-l1.x;
c608     p=intersection(p,c.c,l1,l2);
3855     t=sqrt(c.r*c.r-dis(p,c.c)*dis(p,c.c))/
        dis(l1,l2);
f468     p1.x=p.x+(l2.x-l1.x)*t;
        p1.y=p.y+(l2.y-l1.y)*t;
        p2.x=p.x-(l2.x-l1.x)*t;
        p2.y=p.y-(l2.y-l1.y)*t;
    }
    // 计算圆与圆的交点, 保证圆与圆有交点圆心不
    重合,
void intersection_circle_circle(CP &c1,
    double r1,CP &c2,double r2,point& p1,
    point& p2){
    point u,v;
    double t;
    t=(1+(r1*r1-r2*r2)/dis(c1,c2)/dis(c1,c2
        ))/2;
    u.x=c1.x+(c2.x-c1.x)*t;
    u.y=c1.y+(c2.y-c1.y)*t;
    v.x=u.x+c1.y-c2.y;
    v.y=u.y-c1.x+c2.x;
    intersection_line_circle(c1,r1,u,v,p1,
        p2);
}
void intersection_circle_circle(CC &c1,CC
    &c2,point& p1,point& p2){
    point u,v;
    double t;
    t=(1+(c1.r*c1.r-c2.r*c2.r)/dis(c1.c,c2.
        c)/dis(c1.c,c2.c))/2;
    u.x=c1.c.x+(c2.c.x-c1.c.x)*t;
    u.y=c1.c.y+(c2.c.y-c1.c.y)*t;
    v.x=u.x+c1.c.y-c2.c.y;
    v.y=u.y-c1.c.x+c2.c.x;
    intersection_line_circle(c1.c,c1.r,u,v,
        p1,p2);
}
// 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
bool inside_circle_polygon(CP &c, double
    r, int n, point * polygon){
    if (!inside_polygon(c, n, polygon, 1))
        return false;
    for (int i = 0; i < n; ++i)
        if (disptoline(c,polygon[i], polygon
            [(i + 1) % n]) < r)
            return false;
    return true;
}
bool inside_circle_polygon(CC &c, int n,
    point * polygon){
    if (!inside_polygon(c.c, n, polygon, 1)
        )
        return false;
    for (int i = 0; i < n; ++i)

```

```

a14b     if (disptoline(c.c,polygon[i],
338e         polygon[(i + 1) % n]) < c.r)
3361         return false;
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);
0e20     double angp = acos(r / d);
736d     double ang0 = atan2(p.y - c.y, p.x - c.
        x);
0150     a.x = c.x + r * cos(ang0 + angp);
fc8b     a.y = c.y + r * sin(ang0 + angp);
8b80     b.x = c.x + r * cos(ang0 - angp);
0c1e     b.y = c.y + r * sin(ang0 - angp);
95cf }
6800 void tangent_point_circle(CC &c, CP &p,
        point &a, point &b){
ad5f     double d = dis(c.c, p);
5c2a     double angp = acos(c.r / d);
5922     double ang0 = atan2(p.y - c.c.y, p.x -
        c.c.x);
e086     a.x = c.c.x + c.r * cos(ang0 + angp);
a8d8     a.y = c.c.y + c.r * sin(ang0 + angp);
b6f5     b.x = c.c.x + c.r * cos(ang0 - angp);
aa9f     b.y = c.c.y + c.r * sin(ang0 - angp);
95cf }
427e
427e // 求内切圆, 返回两个切线
0d4d void incut_circle_circle(CP &c1,double r1
        ,CP &c2,double r2,line& l1,line& l2){
b709     double d = sqrt(dis2(c1, c2) - sqr(r1 +
        r2));
cc70     point p1, p2;

intersection_circle_circle(c1, r1 + r2, d070
        c2, d, p1, p2);
11.a = (p1 * r1 + c1 * r2) / (r1 + r2); 6042
11.b = l1.a + (c2 - p1); 6bb3
12.a = (p2 * r1 + c1 * r2) / (r1 + r2); 4a4c
12.b = l2.a + (c2 - p2); ea2d
}求原点原语扇形的夹角 95cf
// 427e
double area_circle_angle(CP &p1, CP &p2, 5ad0
        CP &c, double r){
        double alpha = fabs(atan2(p1.y - c.y,
        p1.x - c.x) - atan2(p2.y - c.y, p2.x
        - c.x));
        if (alpha > pi) alpha = 2 * pi - alpha; 9617
        return alpha / 2 * r * r; 25ea
} 95cf
// 求三角形的的外接圆 427e
void circleoftri(CP &a, CP &b, CP &c, 6ff9
        circle &tmp){
        tmp.c = circumcenter(a,b,c); d028
        tmp.r = dis(a, tmp.c); cdd7
} 95cf
// 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){
        //cout<<n<<endl; 427e
        if (n == 0) tmp.r = -2; 7707
        else if (n == 1){ a7ed
            tmp.c = p[0]; c330
            tmp.r = 0; 0541
        }else if (n == 2){ 02a8
            tmp.r = dis(p[0], p[1]) / 2; fa16
            tmp.c = (p[0] + p[1]) / 2; 8222
        }else if (n == 3) 119b
            circleoftri(p[0], p[1], p[2], tmp); 87a4
} 95cf
void min_circle(int n, point *p, int m, e0ae
        point *down, circle &c){
        min_circle_reduce(m, down, c); 427e
        if (m == 3) return; 425e
        for (int i = 0; i < n; ++i){ 1ae8
            //cout<<i<<" "<<n<<endl; 6c2f
            //cout<<dis(p[i], c.c)<<' '<<c.r<< 427e
                endl;
            if (dis(p[i], c.c) > c.r){ 899e
                //cout<<m<<"yes"<<endl; 427e
                down[m] = p[i]; 03c5
                min_circle(i, p, m + 1, down, c); 85d9
            }
        }
    
```



```

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;
1e19     if ((l1 > r2)&&(l2 > r2)){
98ff         point p3, p4;
fbb4         s = area_circle_angle(p2, p1, c, r);
a4ed         if (disptoseg(c, p1, p2) < r){
4917             intersection_line_circle(c, r, p1,
p2, p3, p4);
7d00             if (dis2(p3, p1) > dis2(p4, p1))
swap(p3, p4);
a289             s -= area_circle_angle(p3, p4, c, r
) - area_triangle(p3, c, p4);
95cf         }
c890         return s * flag;
95cf     }
bcd6     if (l1 < l2){
98ff         point p3, p4;
4917         intersection_line_circle(c, r, p1, p2
, p3, p4);
912e         if (dmult(p3, p2, p1) <= 0) p3 = p4;
f0b8         s = area_triangle(p1, p3, c) +
area_circle_angle(p3, p2, c, r);
c890         return s * flag;
8e2e     }else{
98ff         point p3, p4;
4917         intersection_line_circle(c, r, p1, p2
, p3, p4);
f453         if (dmult(p3, p1, p2) <= 0) p3 = p4;

```

```

s = area_triangle(p2, p3, c) +
area_circle_angle(p3, p1, c, r);
return s * flag;
}
}求圆和多边形的相交面积
//
double area_polygon_circle(int n, point p
[], CP &c, double r){
double ans = 0;
for (int i = 0; i < n; ++i)
ans += area_triangle_circle(c, r, p[i
], p[(i + 1) % n]);
return fabs(ans);
}

```

2.9 网格

```

#define abs(x) ((x)>0?(x):- (x))
struct point{int x,y;};

int gcd(int a,int b){
return b?gcd(b,a%b):a;
}

// 多边形上的网格点个数
int grid_onedge(int n,point* p){
int i,ret=0;
for (i=0;i<n;i++)
ret+=gcd(abs(p[i].x-p[(i+1)%n].x),abs
(p[i].y-p[(i+1)%n].y));
return ret;
}

// 多边形内的网格点个数
int grid_inside(int n,point* p){
int i,ret=0;
for (i=0;i<n;i++)
ret+=p[(i+1)%n].y*(p[i].x-p[(i+2)%n].
x);
return (abs(ret)-grid_onedge(n,p))/2+1;
}

```

2.10 区域中点集个数

```

// 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
// 求xy 到xz 角度里点的个数
int pointinang(int x, int y, int z, int m
, int f[][MAXN], int index[][MAXN]){
if (index[x][z] < index[x][y]) return m
+ f[x][z] - f[x][y] + 1;
return f[x][z] - f[x][y] - 1;
}

```

427e			95cf
427e	// 求三角形xyz 中的点的个数, 输入总的点的个数,	}	95cf
427e	//f[x][y] 表示x 为中心ang 比y 小的点的个数	}	95cf
427e	//h[x][y] 表示xy 左边的点的个数	}	427e
427e	//index[x][y] 表示x 为中心极角排序后的序		427e
e42a	int pointintri(point p[], int x, int y, int z, int m, int f[][MAXN], int h[][MAXN], int index[][MAXN]){	// 求p1 中任意三点切割的七个区域的p2 点集的个数, 要求三点不共线	427e
283b	if (xmult(p[z], p[y], p[x]) > 0) swap(y, z);	// 求xy 到xz 角度里点的个数	427e
53ef	int a = h[x][z] + h[y][x] + h[z][y];	int pointinang2(int x, int y, int z, int m, int f[][MAXN], int index[][MAXN]){	251e
5a59	a += pointinang(x, y, z, m, f, index);	if (index[x][z] < index[x][y]) return m + f[x][z] - f[x][y];	35b8
f5d7	a += pointinang(y, z, x, m, f, index);	return f[x][z] - f[x][y];	bd1e
150c	a += pointinang(z, x, y, m, f, index);	}	95cf
fd8e	a -= 2 * m;		427e
5ffd	return a;	// 求三角形xyz 中的点的个数, 输入总的点的个数,	427e
95cf	}	//f[x][y] 表示x 为中心ang 比y 小的点的个数	427e
427e		//h[x][y] 表示xy 左边的点的个数	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	//index[x][y] 表示x 为中心极角排序后的序	427e
427e	//f[x][y] 表示x 为中心ang 比y 小的点的个数	int pointintri2(point p[], int x, int y, int z, int m, int f[][MAXN], int h[][MAXN], int index[][MAXN]){	73ff
427e	//h[x][y] 表示xy 左边的点的个数	if (xmult(p[z], p[y], p[x]) > 0) swap(y, z);	283b
427e	//index[x][y] 表示x 为中心极角排序后的序	int a = h[x][z] + h[y][x] + h[z][y];	53ef
614b	void pointinarea(int n, point p1[], int f[][MAXN], int h[][MAXN], int index[][MAXN]){	a += pointinang2(x, y, z, m, f, index);	2851
60da	point p[2 * MAXN];	a += pointinang2(y, z, x, m, f, index);	61bc
6c2f	for (int i = 0; i < n; ++i){	a += pointinang2(z, x, y, m, f, index);	164e
8abb	int cnt = 0;	a -= 2 * m;	fd8e
fde8	for (int j = 0; j < n; ++j)	return a;	5ffd
4a23	if (i != j){	}	95cf
ca58	p[cnt] = p1[j];		427e
28c2	p[cnt].index = j;	// 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
2f66	p[cnt++].ang = atan2(p1[j].y - p1[i].y, p1[j].x - p1[i].x);	//f[x][y] 表示x 为中心ang 比y 小的点的个数	427e
95cf	}	//h[x][y] 表示xy 左边的点的个数	427e
b073	sort(p, p + cnt, cmp);	//index[x][y] 表示x 为中心极角排序后的序	427e
07de	for (int j = 0; j < cnt; ++j){	void pointinarea2(int n, point p1[], int m, point p2[], int f[][MAXN], int h[][MAXN], int index[][MAXN]){	3cbd
fcfa	p[j + cnt] = p[j];	point p[2 * (MAXN + MAXM)];	9941
ceb9	p[j + cnt].ang += 2 * pi;	for (int i = 0; i < n; ++i){	6c2f
95cf	}	int cnt = 0;	8abb
07b3	for (int k = 0, j = 0, mine = 0, l = 0; k < cnt; ++k){	for (int j = 0; j < n; ++j)	fde8
9b9c	while (p[j].ang - p[k].ang < pi){	if (i != j){	4a23
ae6d	++mine;	p[cnt] = p1[j];	ca58
917f	++j;	p[cnt].index = j;	28c2
95cf	}	p[cnt++].ang = atan2(p1[j].y - p1[i].y, p1[j].x - p1[i].x);	2f66
5021	--mine;	}	95cf
ca69	h[i][p[k].index] = mine;	for (int j = 0; j < m; ++j){	6613
c92b	f[i][p[k].index] = 1;	p[cnt] = p2[j];	acb5
e5d0	index[i][p[k].index] = k;		
713f	++l;		

```

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] = 1;
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

```

652e #define eps 1e-8
c1b0 #define fabs(x) ((x) > 0? (x): -(x))
0102 #define zero(x) (fabs(x) < eps)
dca2 #define sqr(x) ((x)*(x))
12d8 #define _sign(x) ((x)>eps?1:((x)<=-eps
          ?2:0))
13f1 const double pi = acos(-1);
427e
427e // 点的定义
b6b2 struct point3{
9d7e     double x, y, z;
6c19     point3(){x = 0;y = 0;z = 0;}
18ee     point3(double sx, double sy, double sz)
          {
e87b         x = sx;
d22b         y = sy;
826b         z = sz;
95cf     }
548e     bool operator <(const point3 &b)const{
1737         if (b.x == x){
e65c             if (y == b.y) return z < b.z;

```

```

          return y < b.y;
          }
          return x < b.x;
          }
point3 operator - (const point3 &b)
const {
point3 a;
a.x = x - b.x;
a.y = y - b.y;
a.z = z - b.z;
return a;
}
point3 operator + (const point3 &b)
const{
point3 a;
a.x = x + b.x;
a.y = y + b.y;
a.z = z + b.z;
return a;
}
point3 operator / (const double &c)
const{
point3 a;
a.x = x / c;
a.y = y / c;
a.z = z / c;
return a;
}
point3 operator * (const double &c)
const{
point3 a;
a.x = x * c;
a.y = y * c;
a.z = z * c;
return a;
}
bool operator == (const point3 &p)
const {
return zero(x - p.x)&&zero(y - p.y)&&
zero(z - p.z);
}
friend ostream& operator << (ostream &
out, const point3 &a);
};
ostream& operator << (ostream &out, const
point3 &a){
out<<a.x<<' '<<a.y<<' '<<a.z;
return out;
}
// 边定义
struct line3{
point3 a, b;

```

```

2f79     line3(){};
64c6     line3(const point3 &p1, const point3 &
0fa8         p2){
ce41         a = p1;
95cf         b = p2;
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
427e // 计算dot product U . V
f42e double dmult(const point3 &u, const
        point3 &v){
49f2     return u.x * v.x + u.y * v.y + u.z * v.
        z;
95cf }
427e
427e // 取平面法向量
388c point3 pvec(const plane3 &s){
07d5     return xmult(s.a - s.b, s.b - s.c);
95cf }
afb0 point3 pvec(const point3 &s1, const
        point3 &s2, const point3 &s3){
fd86     return xmult(s1 - s2, s2 - s3);
95cf }
427e
427e // 两点距离, 单参数取向量大小
9e30 double dis(const point3 &p1, const point3
        &p2){
fafb     return sqrt(sqr(p1.x - p2.x) + sqr(p1.y
        - p2.y) + sqr(p1.z - p2.z));
95cf }
be8b double dis2(const point3 &p1, const
        point3 &p2){
        return sqrt(p1.x - p2.x) + sqr(p1.y - p2
        .y) + sqr(p1.z - p2.z);
        }
        // 向量大小
        double len(const point3 &p){
            return sqrt(sqr(p.x) + sqr(p.y) + sqr(p
            .z));
        }
        // 判三点共线
        int dots_inline(const point3 &p1, const
        point3 &p2, const point3 &p3){
            return len(xmult(p1 - p2, p2 - p3)) <
            eps;
        }
        // 判四点共面
        int dots_onplane(const point3 &a, const
        point3 &b, const point3 &c, const
        point3 &d){
            return zero(dmult(pvec(a, b, c), d - a)
            );
        }
        // 判点是否在线段上, 包括端点和共线
        int dot_online_in(const point3 &p, const
        line3 &l){
            return zero(len(xmult(p - l.a, p - l.b)
            ))&&
            (l.a.x - p.x) * (l.b.x - p.x) < eps&&
            (l.a.y - p.y) * (l.b.y - p.y) < eps&&
            (l.a.z - p.z) * (l.b.z - p.z) < eps;
        }
        int dot_online_in(const point3 &p, const
        point3 &l1, const point3 &l2){
            return zero(len(xmult(p - l1, p - l2)))
            &&
            (l1.x - p.x) * (l2.x - p.x) < eps&&
            (l1.y - p.y) * (l2.y - p.y) < eps&&
            (l1.z - p.z) * (l2.z - p.z) < eps;
        }
        // 判点是否在线段上, 不包括端点
        int dot_online_ex(const point3 &p, const
        line3 &l){
            return dot_online_in(p, l) && !(p == l
            .a)&&!(p == l.b));
        }
        int dot_online_ex(const point3 &p, const
        point3 &l1, const point3 &l2){
            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)) -
febfb     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
int opposite_side(const point3 &p1, const
    point3 &p2, const line3 &l){
    return dmult(xmult(l.a - l.b, p1 - l.b)
        , xmult(l.a - l.b, p2 - l.b)) < -eps;
}
int opposite_side(const point3 &p1, const
    point3 &p2, const point3 &l1, const
    point3 &l2){
    return dmult(xmult(l1 - l2, p1 - l2),
        xmult(l1 - l2, p2 - l2)) < -eps;
}
// 判两点在平面同侧, 点在平面上返回0
int same_side(const point3 &p1, const
    point3 &p2, const plane3 &s){
    return dmult(pvec(s), p1 - s.a) * dmult
        (pvec(s), p2 - s.a) > eps;
}
int same_side(const point3 &p1, const
    point3 &p2, const point3 &s1, const
    point3 &s2, const point3 &s3){
    return dmult(pvec(s1, s2, s3), p1 - s1) *
        dmult(pvec(s1, s2, s3), p2 - s1) > eps
        ;
}
// 判两点在平面异侧, 点在平面上返回0
int opposite_side(const point3 &p1, const
    point3 &p2, const plane3 &s){
    return dmult(pvec(s), p1 - s.a) * dmult
        (pvec(s), p2 - s.a) < -eps;
}
int opposite_side(const point3 &p1, const
    point3 &p2, const point3 &s1, const
    point3 &s2, const point3 &s3){
    return dmult(pvec(s1, s2, s3), p1 - s1) *
        dmult(pvec(s1, s2, s3), p2 - s1) < -
        eps;
}
// 判两直线平行
int parallel(const line3 &u, const line3
    &v){
    return len(xmult(u.a - u.b, v.a - v.b))
        < eps;
}
int parallel(const point3 &u1, const
    point3 &u2, const point3 &v1, const
    point3 &v2){
    return len(xmult(u1 - u2, v1 - v2)) <
        eps;
}

```

```

427e // 判两平面平行
3062 int parallel(const plane3 &u, const
c6ae plane3 &v){
95cf return len(xmult(pvec(u) , pvec(v))) <
7d09 eps;
ef70 }
95cf int parallel(const point3 &u1, const
427e point3 &u2, const point3 &u3, const
427e point3 &v1, point3 v2, point3 v3){
454a return len(xmult(pvec(u1, u2, u3), pvec
31d4 (v1, v2, v3))) < eps;
95cf }
06d0 int parallel(const point3 &l1, const
d311 point3 &l2, const point3 &s1, const
95cf point3 &s2, const point3 &s3){
427e return zero(dmult(l1 - l2, pvec(s1, s2,
427e s3)));
b98a }
85bf // 判两直线垂直
b77b int perpendicular(const line3 &u, const
4547 line3 &v){
95cf return zero(dmult(u.a - u.b, v.a - v.b)
427e );
427e }
0e75 int perpendicular(const plane3 &u, const
da40 plane3 &v){
95cf return zero(dmult(pvec(u), pvec(v)));
75f2 }
8919 int perpendicular(const point3 &u1, const
95cf point3 &u2, const point3 &u3, const
427e point3 &v1, const point3 &v2, const
427e point3 &v3){
16f9 return zero(dmult(pvec(u1, u2, u3),
pvec(v1, v2, v3)));
plane3 &s){
return len(xmult(l.a - l.b, pvec(s))) <
eps;
}
int perpendicular(const point3 &l1, const
point3 &l2, const point3 &s1, const
point3 &s2, const point3 &s3){
return len(xmult(l1 - l2, pvec(s1, s2,
s3))) < eps;
}
// 判两线段相交, 包括端点和部分重合
int intersect_in(const line3 &u, const
line3 &v){
if (!dots_onplane(u.a, u.b, v.a, v.b))
return 0;
if (!dots_inline(u.a, u.b, v.a) ||
dots_inline(u.a, u.b, v.b))
return !same_side(u.a, u.b, v)&&!
same_side(v.a, v.b, u);
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);
}
int intersect_in(const point3 &u1, const
point3 &u2, const point3 &v1, const
point3 &v2){
if (!dots_onplane(u1, u2, v1, v2))
return 0;
if (!dots_inline(u1, u2, v1) ||
dots_inline(u1, u2, v2))
return !same_side(u1, u2, v1, v2)&&!
same_side(v1, v2, u1, u2);
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);
}
// 判两线段相交, 不包括端点和部分重合
int intersect_ex(const line3 &u, const
line3 &v){
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);
}
int intersect_ex(const point3 &u1, const
point3 &u2, const point3 &v1, const
point3 &v2){
return dots_onplane(u1, u2, v1, v2)&&
opposite_side(u1, u2, v1, v2)&&
opposite_side(v1, v2, u1, u2);
}

```

```

427e // 判线段与空间三角形相交, 包括交于边界
427e 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 // 判线段与空间三角形相交, 不包括交于边界
427e 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 }
ebd9 int intersect_ex(const point3 &l1, const
point3 &l2, const point3 &s1, const
point3 &s2, const point3 &s3){
3e15     return opposite_side(l1,l2,s1,s2,s3)&&
opposite_side(s1,s2,l1,l2,s3)&&
ebb0     opposite_side(s2,s3,l1,l2,s1)&&
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 !
427e // 线段交点请另外判线段相交同时还是要判断是否
平行 ( ! )
58cf point3 intersection(const line3 &u, const
line3 &v){
87cc     point3 ret=u.a;
273a     double t=((u.a.x-v.a.x)*(v.a.y-v.b.y)-(
u.a.y-v.a.y)*(v.a.x-v.b.x))/
9cb3         (((u.a.x-u.b.x)*(v.a.y-v.b.y)-(u.a.
y-u.b.y)*(v.a.x-v.b.x)));
1143     ret.x+=(u.b.x-u.a.x)*t;
12e9     ret.y+=(u.b.y-u.a.y)*t;
1037     ret.z+=(u.b.z-u.a.z)*t;
ee0f     return ret;
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
-v1.y)*(v1.x-v2.x))                        a7db
/(((u1.x-u2.x)*(v1.y-v2.y)-(u1.y-u2.
y)*(v1.x-v2.x)));                          16f0
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
// 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.
a.y-l.a.y)+ret.z*(s.a.z-l.a.z))/          f84a
(ret.x*(l.b.x-l.a.x)+ret.y*(l.b.y-l.a
.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
point3 intersection(const point3 &l1,      2ec4
const point3 &l2, const point3 &s1,
const point3 &s2, const point3 &s3){
point3 ret=pvec(s1,s2,s3);                    2f24
double t=(ret.x*(s1.x-l1.x)+ret.y*(s1.y
-l1.y)+ret.z*(s1.z-l1.z))/                38b0
(ret.x*(l2.x-l1.x)+ret.y*(l2.y-l1.y)+
ret.z*(l2.z-l1.z));
ret.x=l1.x+(l2.x-l1.x)*t;                    6302
ret.y=l1.y+(l2.y-l1.y)*t;                    9cd9
ret.z=l1.z+(l2.z-l1.z)*t;                    29b2
return ret;                                  ee0f
}                                              95cf
// 计算两平面交线, 注意事先判断是否平行, 并      427e
保证三点不共线 !
line3 intersection(const plane3 &u, const  69e4
plane3 &v){
line3 ret;                                    57af
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);
ret.b=parallel(v.c,v.a,u.a,u.b,u.c)?      033e
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    // 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
7ed7    double ptoplane(const point3 &p, const
        plane3 &s){
b87b        return fabs(dmult(pvec(s), p - s.a))/
            len(pvec(s));
95cf    }
33d1    double ptoplane(const point3 &p, const
        point3 &s1, const point3 &s2, const
        point3 &s3){
3cea        return fabs(dmult(pvec(s1,s2,s3), p -
            s1))/len(pvec(s1,s2,s3));
95cf    }
427e    // 直线到直线距离
eaed    double linetoline(const line3 &u, const
        line3 &v){
79e9        point3 n=xmult(u.a - u.b, v.a - v.b);
2a04        return fabs(dmult(u.a - v.a,n))/len(n);
95cf    }
c288    double linetoline(const point3 &u1, const
        point3 &u2, const point3 &v1, const
        point3 &v2){
a9a1        point3 n=xmult(u1 - u2, v1 - v2);
04de        return fabs(dmult(u1 - v1,n))/len(n);
    }
    // 两直线夹角cos 值
double angle_cos(const line3 &u, const
    line3 &v){
    return dmult(u.a - u.b, v.a - v.b)/len(
        u.a - u.b)/len(v.a - v.b);
    }
double angle_cos(const point3 &u1, const
    point3 &u2, const point3 &v1, const
    point3 &v2){
    return dmult(u1 - u2, v1 - v2)/len(u1 -
        u2)/len(v1 - v2);
    }
    // 两平面夹角cos 值
double angle_cos(const plane3 &u, const
    plane3 &v){
    return dmult(pvec(u),pvec(v))/len(pvec(
        u))/len(pvec(v));
    }
double angle_cos(const point3 &u1, const
    point3 &u2, const point3 &u3, const
    point3 &v1, const point3 &v2, const
    point3 &v3){
    return dmult(pvec(u1,u2,u3),pvec(v1,v2,
        v3))/len(pvec(u1,u2,u3))/len(pvec(v1,
        v2,v3));
    }
    // 直线平面夹角sin 值
double angle_sin(const line3 &l, const
    plane3 &s){
    return dmult(l.a - l.b,pvec(s))/len(l.a
        - l.b)/len(pvec(s));
    }
double angle_sin(const point3 &l1,const
    point3 &l2, const point3 &s1, const
    point3 &s2, const point3 &s3){
    return dmult(l1 - l2,pvec(s1,s2,s3))/
        len(l1 - l2)/len(pvec(s1,s2,s3));
    }
3.3 面积
    // 求三角形有向表面积, 输入三个顶点
double area_triangle(const plane3 &p){
    return len(xmult(p.b - p.a, p.c - p.a))
        /2;
    }
double area_triangle(const point3 &p1,
    const point3 &p2, const point3 &p3){
    return len(xmult(p2 - p1, p3 - p1))/2;
    }

```



```

6e8b double area_triangle(const point3 &p2,
c28a     const point3 &p3){
95cf     return len(xmult(p2, p3))/2;
427e }
2eca // 求多边形有向表面积, 输入三个顶点
99ec double area_polygon(int n, point3* p){
85c3     double s;
d161     for (int i = 0; i < n; ++i)
        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 }
2cde double volume_tetrahedron(const plane3 &
p){
9493     return dmult(xmult(p.a, p.b), p.c) / 6;
95cf }
427e // 求多面体有向体积
0532 double volume_polygon(int n, plane3 *
polygon){
0ea9     double c = 0;
85c3     for (int i = 0; i < n; ++i)
8088         c += volume_tetrahedron(polygon[i]);
14df     return c;
95cf }

```

3.5 重心

```

427e // 三角形重心
4ce9 point3 barycenter(const point3 &a, const
point3 &b, const point3 &c){
85a3     return a + b + c / 3;
95cf }
427e // 四面体重心
6485 point3 barycenter(const point3 &a, const
point3 &b, const point3 &c, const
point3 &d){
3cb1     return (a + b) + (c + d) / 4;
95cf }
427e // 多面体重心

```

```

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    e51b
[i]);
v += j;                                58e1
c = c + (polygon[i].a + polygon[i].b    32b4
+ polygon[i].c) * j;
}
return c / (4 * v);                      95cf
}                                         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[MAXN];                                  4cef
int edge[MAXN][MAXN];                     91d4
int tot;                                   8164
int next(int x){                           0019
if (s[x].next == x) return x;             d010
return s[x].next = next(s[x].next);       901a
}                                           95cf
void add(int a, int b, int c, point3 *p){ f01b
s[tot].p[0] = a;                          582f
s[tot].p[1] = b;                          661d
s[tot].p[2] = c;                          50e3
s[tot].p[3] = a;                          c4b1
s[tot].f = xmult(p[b] - p[a], p[c] - p[  202c
a]);
s[tot].out = false;                       7fb3
for (int i = 0; i < 3; ++i)               100b
edge[s[tot].p[i]][s[tot].p[i + 1]] =    1e79
tot;
++tot;                                    ac2d
}                                           95cf
void add(int a, int b, int c, int d,      c1c6
point3 *p){
point3 f = xmult(p[b] - p[a], p[c] - p[  05a3
a]);
if (dmult(f, p[d] - p[a]) > 0) add(a, c  40d9
, b, p);
else add(a, b, c, p);                    3c28
}                                           95cf
// 主程序输入顶点个数, 点集, 返回面,    427e
// 要求不是所有点共面                    427e
int get_convex(int n, point3* p, plane3* 4db6

```

```

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)
1807                       add(s[j].p[k], s[j].p[k + 1],
                           i, p);
68c3           s[j].next = j + 1;
95cf       }
95cf   }
576f   int i, j;
d4d9   for (i = 0, j = next(0); j < tot; ++i,
        j = next(j + 1)){
82ac       convex[i].a = p[s[j].p[0]];
c1d7       convex[i].b = p[s[j].p[1]];
f7bf       convex[i].c = p[s[j].p[2]];
95cf   }
ffec   return i;
95cf }

```

4 数论

4.1 头文件

```

427e // independent
e0a5 #include<iostream>
54ff #include<algorithm>
c928 #include<cmath>
59b9 #include<cstdio>
59b9 #include<cstdint>
8c52 #include<map>
6326 #include<set>

```

```

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

```

4.3 线性筛法

```

// independent                                427e
const int MAXN=10000000;                      62a8
int cprime[MAXN],used=0;                      1858
char p[MAXN];                                  8a94
void prime_(){                                  0919

```

```

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*cpime[j]>MAXN) break;
ece6            p[i*cpime[j]]=true;
0f27            if(i%cpime[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:  $a*x = b \bmod m$  的所有
      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;
89a5     for(int i=0;i<len;i++) ans[i]=(base+i
*(m/len))%m;
3361     return true;
95cf }

```

4.5 中国剩余定理

```

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

```

4.6 离散对数

```

// independent
// 求解形如  $a^x = b \bmod MOD$  的方程, , , 已
      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(flgl[k] != idx){
fcdc         flgl[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     Hash[top].a = a;
d291     Hash[top].b = b;
ad12 }
95cf
int find(int b){
99f4     int k = b & maxn;
2e3c     if(flgl[k] != idx) return -1;
a4dc     while(k != -1){
9030         if(Hash[k].b == b) return Hash[k].a;
b35f         k = Hash[k].next;
5551     }
95cf     return -1;
fb5e }
95cf
int gcd(int a,int b){return b?gcd(b,a%b):
e8bb 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
int pow_mod(LL a,int b,int c)
49b5 {
4506     LL ret=1%c;a%=c;
8fef     while(b)
7c06     {
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     }
fea4     int M=(int)ceil(sqrt((double)C));
2662     for(buf=1%C,i=0;i<=M;buf=buf*A%C,++i)
        ins(i,buf);
21a3     for(i=0,K=pow_mod((LL)A,M,C);i<=M;D=D*K
        %C,++i){
5d4e         tmp=Inval((int)D,B,C);int w ;
b3d8         if(tmp>0&&(w = find(tmp)) != -1)
            return i*M+w+d;
95cf     }
fb5e     return -1;
95cf }
427e
427e
427e
3117 int main(){
1e17     int A,B,C;
2072     while(scanf("%d%d%d",&A,&C,&B)!=EOF,A
        || B || C){
1fc4         B %= C;
1719         int tmp=BabyStep(A,B,C);
6a67         if(tmp<0)puts("No Solution");else
            printf("%d\n",tmp);
95cf     }
7021     return 0;
95cf }

```

4.7 MillerRabin

```

427e // depend on 基础: multiMod

```

```

bool witness ( LL s , LL n ) {
    LL u = n - 1 ;
    int t = 0 ;
    while ( ( u & 1 ) == 0 ) u >>= 1 , t ++
    ;
    LL x = FM ( s , u , n ) ;
    while ( t -- ) {
        LL tmp = x ;
        x = multiMod ( x , x , n ) ;
        if ( x == 1 ) {
            if ( tmp == n - 1 || tmp == 1 )
                return false ;//may be prime
            else return true ;//composite
        }
    }
    return true ; //composite
}

bool millerRabin ( LL n , const int times
    = 3 ) {
    if ( n == 2 ) return true ;
    if ( ( n & 1 ) == 0 || n < 2 ) return
        false ;
    int i = times ;
    while ( i -- ) {
        LL s = rand ( ) % ( n - 1 ) + 1 ;
        if ( witness ( s , n ) ) return false
        ;
    }
    return true ;
}

```

4.8 PollardRho

```

// depend on 基础: , multiModFM
LL multiMod ( LL a , LL b , LL n ) {
    a %= n ;
    b %= n ;
    LL s = 0 ;
    while( b ) {
        if( b & 1 ) {
            s += a ;
            if( s >= n ) s -= n ;
        }
        a <<= 1 ; b >>= 1 ;
        if(a >= n ) a -= n ;
    }
    return s ;
}

LL FM ( LL s , LL u , LL n ) {
    s %= n ;
    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     }
3361     return true ; //composite
95cf }
427e
2daf bool millerRabin ( LL n , const int times
    = 3 ) {
89e8     if ( n == 2 ) return true ;
62ac     if ( ( n & 1 ) == 0 || n < 2 ) return
        false ;
d8c4     int i = times ;
148a     while ( i — ) {
2e40         LL s = rand ( ) % ( n - 1 ) + 1 ;
6474         if ( witness ( s , n ) ) return false
            ;
95cf     }
3361     return true ;
95cf }
4990 LL gcd ( LL a , LL b ) {
22aa     if ( b == 0 ) return a ;
7b09     return gcd ( b , a % b ) ;
95cf }
32db LL pollard_rho ( LL n ) {
61ac     LL x , y , k , d ;
2cce     x = y = rand ( ) % n ;
7924     k = 2 ;
0d80     int i = 1 ;
25c7     int c = rand ( ) % n ;
427e
1026     while ( true ) {

```

```

        i ++ ;
        x = ( multiMod ( x , x , n ) + c ) %
            n ;
        if ( y == x ) return 1 ; //restart
        else if ( y > x ) d = gcd ( y - x , n
            ) ;
        else d = gcd ( x - y , n ) ;
        if ( d != 1 && d != n - 1 ) return d
            ;
        else {
            if ( i == k ) {
                y = x ;
                k <= 1 ;
            }
        }
    }
}存放分解出的质因子
//
LL factors [ 54 ] ;初始化为
//0
int cnt ;
void split ( LL n ) { //n != 1
    if ( millerRabin ( n ) ) factors [ cnt
        ++ ] = n ;
    else {
        LL p ;
        do {
            p = pollard_rho ( n ) ;
        } while ( p == n || p == 1 ) ;
        split ( p ) ;
        split ( n / p ) ;
    }
}

```

4.9 矩阵基础

```

// independent
struct Matrix{int m[MAXN][MAXN],l,r;
    Matrix(int w){
        l=r=w;
        memset(m,0,sizeof m);
    }
    Matrix(){l=r=0;memset(m,0,sizeof m);}
};
// a.r = b.l
Matrix operator * (Matrix a,Matrix b){
    Matrix c;
    memset(c.m,0,sizeof c.m);
    c.l=a.l,c.r=b.r;
    for(int i=0;i<a.l;i++){
        for(int j=0;j<b.r;j++){
            for(int k=0;k<a.r;k++){
                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 // 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){
e385         if(t&1) ans=ans*mid;
c1cb         mid=mid*mid;
2f01         t>>=1;
95cf     }
4206     return ans;
95cf }

```

4.10 高斯消元

```

427e // independent
fb02 #define MAXN 100
c1b0 #define fabs(x) ((x)>0?(x):- (x))
b76f #define eps 1e-10列主元
427e
427e //消去求解gaussa[][]x[]=b[]返回是否有唯一解
      若有解在
427e //,b中[]
f2e9 int gauss_cpivot(int n,double a[][MAXN],
      double b[]){
c75e     int i,j,k,row;
ea2e     double maxp,t;
ab8e     for (k=0;k<n;k++){
a1ed         for (maxp=0,i=k;i<n;i++)
8dd1             if (fabs(a[i][k])>fabs(maxp))
f0ed                 maxp=a[i][k];
ff5d             if (fabs(maxp)<eps)
7021                 return 0;
7dcf             if (row!=k){
0dff                 for (j=k;j<n;j++)

```

```

      t=a[k][j],a[k][j]=a[row][j],a[row
      ][j]=t;
      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>
#include<cstring>
#include<stdlib>
59b9
ef2f
bffa
427e
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
      int sz;
4a30
427e
      void init()
88f1
      {
4506
          for (int i=0;i<=sz;i++) S[i].s=S[i].c
bd30
              =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);
9a2e    x->max=MAX(x->max, x->C[1]->ml+x->s);
8176    x->max=MAX(x->max, x->C[0]->mr+x->s+x
        ->C[1]->ml);
0181    x->ml=MAX(x->C[0]->ml, x->C[0]->tot+x
        ->s);
7a57    x->ml=MAX(x->ml, x->C[0]->tot+x->s+x->
        C[1]->ml);
77d1    x->mr=MAX(x->C[1]->mr, x->C[1]->tot+x
        ->s);
a71b    x->mr=MAX(x->mr, x->C[1]->tot+x->s+x->
        C[0]->mr);*/
95cf    }
427e

```

```

/*void labledown( SplayNode *x){
    SplayNode *ts;
    int tmp;
    if (x==null||!(x->same||x->rev))
        return ;
    if (x->same){
        x->C[1]->same=x->C[0]->same=1;
        x->C[1]->s=x->C[0]->s=x->s;
        x->tot=x->s*x->sz;
        x->max=x->ml=x->mr=x->tot;
        if (x->s<0)
            x->max=x->ml=x->mr=x->s;
    }
    if (x->rev){
        tmp=x->ml;x->ml=x->mr;x->mr=tmp;
        ts=x->C[1];x->C[1]=x->C[0];x->C[0]=
            ts;
        x->C[1]->rev=!x->C[1]->rev;x->C
            [0]->rev=!x->C[0]->rev;
    }
    x->same=x->rev=0;
}*/
void route( SplayNode *k1, int c){
    SplayNode *k2=k1->f;
    //labledown(k2->C[!c]);labledown(k1->
        C[0]);labledown(k1->C[1]);
    k2->C[c]=k1->C[!c];
    k2->C[c]->f=k2;
    k1->f=k2->f;
    if (k2->f->C[0]==k2) k2->f->C[0]=k1;
    else k2->f->C[1]=k1;
    k2->f=k1;
    k1->C[!c]=k2;
    update(k2);//update(k1);
    if (root==k2) root=k1;
}
SplayNode * rank( int k){
    SplayNode *ts=root;
    int tmp;
    while (k){
        //labledown(ts);
        tmp=ts->C[0]->sz;
        if (k<=tmp) ts=ts->C[0];
        else if (k<=tmp+ts->c+1) break;
        else k-=tmp+ts->c+1,ts=ts->C[1];
    }
    return ts;
}
/*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);
4f2d                 return ;
95cf             }
35a1         else if (k<ts->s) ts=ts->C[0];
b87e         else ts=ts->C[1];
95cf     }
4efc     if (k<ls->s)
4506     {
7728         ls->C[0]=NewNode(ls,k);
96ed         update(ls->C[0]);update(ls);
4b8a         splay(ls->C[0],null);
d268     } else
4506     {
703f         ls->C[1]=NewNode(ls,k);
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 //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     root->f=null;
d16b     root->C[1]=ts->C[1];
469b     ts->C[1]->f=root;
1105     update(root);
d657 }
95cf
427e int find( int k)
d16f {
4506     splay(rank(k+1),null);
d827     return root->s;
1ff2 }
95cf
427e int find_v( int k)
8c3f
```



```

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>
bffa #include<cstdlib>
54ff #include<algorithm>
acb9 #include<queue>
427e
421c using namespace std;
427e
baf7 priority_queue<int> MinNumber;
427e
b47b #define MAXN 300010
06a1 #define INF 1ll<<62
5da6 #define MAX(a,b) ((a)>(b)?(a):(b))
427e
427e
427e
f7cd struct SplayNode{
b275     SplayNode *f,*C[2];
427e     //int s,ml,mr,max,tot,sz;
880b     long long s,c,sz,tot;
9673     int neg,pos,sta;
427e     //bool rev,same;
329b };
427e
e929 int N,st,it;
6c85 char s[100];
437d SplayNode *A[MAXN],*B[MAXN];
427e
427e
11d0 struct SplayTree{
427e

```

```

4a30 int sz;
ea48 SplayNode S[MAXN],*root,*null,*tr;
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;
6b1f     null->tot=0;
0380     null->neg=null->pos=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);
a562     root->neg=root->pos=root->C[1]->neg=
        root->C[0]->pos=root->sta=root->C
        [1]->sta=0;
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;
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
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];
f1b2         else if (k<=tmp+1) break;
fefe         else k-=tmp+1,ts=ts->C[1];
95cf     }
dd0c     return ts;
95cf }
427e
427e
f257 long long query( SplayNode *l,
SplayNode *r)
4506 {
da8a     splay(l,null);splay(r,root);
7602     return root->C[1]->C[0]->tot+root->s+
root->C[1]->s;
95cf }
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){
if (x->f->C[0]==x)
route(x,0);
else
route(x,1);
} else if (x->f->f->C[0]==x->f){
if (x->f->C[0]==x)
route(x->f,0),route(x,0);
else
route(x,1),route(x,0);
} else {
if (x->f->C[1]==x)
route(x->f,1),route(x,1);
else
route(x,0),route(x,1);
}
}
update(x);
}
//rank k
427e
void del( SplayNode *ss)
aae1 {
4506     splay(ss,null);
fc5e     splay(rank(root->C[0]->sz),root);
79ab     SplayNode *ts=root;
56b2     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 }
95cf
427e
void ins( int k, int s)
0edf {
4506     splay(rank(k+1),null);
d827     //splay(rank(k),root);
427e     SplayNode *ts=NewNode(root,s);
e7e2     root->C[0]->f=ts;
7235     ts->C[0]=root->C[0];
3fe8     root->C[0]=ts;
d5f0     A[s]=ts;
f9af     splay(ts,null);
530f }
95cf
void find( int t)
a228 {
4506     SplayNode *ts=root,*ls;
af72     while (ts!=null)
eb9a     {
4506         ls=ts;
17d7         if (ts->C[1]->neg>=t) ts=ts->C[1];
eec3         else if (ts->C[1]->neg+(ts->sta
20e3             <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
299c    int main()
4506    {
1008        int cases=0;
b889        while (scanf("%d\n",&N)!=EOF)
4506        {
db73            T.init();
19f9            while (MinNumber.size()) MinNumber.
                pop();
6b7c            int nown=0,t;
524a            printf("Case #d:\n",++cases);
720b            while (N--)
4506            {
860a                scanf("%s %d\n",s,&t);
b81b                if (s[0]=='i')
4506                {
5006                    t++;
7309                    if (MinNumber.empty()) it=++nown;
c2a0                    else it=MinNumber.top(),
                        MinNumber.pop();
a3af                    ins(t,it);
603e                } else if (s[0]=='r')
4506                {
3834                    MinNumber.push(-t);
bf62                    T.del(A[t]);
bb40                    T.del(B[t]);
b72a                } else if (s[0]=='q')
4506                {
0016                    printf("%I64d\n",T.query(A[t],B[t]

```

```

        ]));
    }
    }
}
}
95cf
95cf
95cf
95cf

5.2 SplayTree

#include<cstdio>
#include<cstring>
#include<cstdlib>

using namespace std;

#define MAXN 40010
#define INF 1ll<<62
#define MAX(a,b) ((a)>(b)?(a):(b))
struct SplayTree{

    struct SplayNode{
        SplayNode *f,*C[2];
        //int s,ml,mr,max,tot,sz;
        long long s,c,sz;
        //bool rev,same;
    } S[MAXN],*root,*null,*tr;

    int sz;

    void init()
    {
        for (int i=0;i<=sz;i++) S[i].s=S[i].c
            =S[i].sz=0;
        sz=0;
        null=NewNode(null,-INF);
        null->s=0;
        null->f=null;
        null->sz=0;
        null->C[0]=null->C[1]=null;
        root=NewNode(null,-INF);
        root->C[1]=NewNode(root,INF);
        update(root->C[1]);
        update(root);
    }

    SplayNode * NewNode( SplayNode *f, long
        long s){
        SplayNode *ts;
        ts=S+ ++sz;
        ts->f=f;
        ts->c=0;
        ts->C[0]=ts->C[1]=null;
        ts->s=s;
        //ts->tot=ts->max=ts->ml=ts->mr=ts->s
        ;
95cf
ef2f
bffa
427e
421c
427e
1c79
06a1
5da6
11d0
427e
f7cd
b275
427e
440f
427e
b01b
427e
4a30
427e
88f1
4506
bd30
1bb9
47f6
ab2e
f607
c969
e3e8
b6b3
0e03
2753
d657
95cf
427e
b77a
dbed
40e5
2d59
bc22
577f
ca6f
427e

```

```

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);
9a2e      x->max=MAX(x->max, x->C[1]->m1+x->s);
8176      x->max=MAX(x->max, x->C[0]->mr+x->s+x
          ->C[1]->m1);
0181      x->m1=MAX(x->C[0]->m1, x->C[0]->tot+x
          ->s);
7a57      x->m1=MAX(x->m1, x->C[0]->tot+x->s+x->
          C[1]->m1);
77d1      x->mr=MAX(x->C[1]->mr, x->C[1]->tot+x
          ->s);
a71b      x->mr=MAX(x->mr, x->C[1]->tot+x->s+x->
          C[0]->mr);*/
95cf    }
427e
1455    /*void labledown( SplayNode *x){
dbed      SplayNode *ts;
6eb3      int tmp;
45c9      if (x==null||!(x->same||x->rev))
          return ;
d66a      if (x->same){
d26a        x->C[1]->same=x->C[0]->same=1;
e573        x->C[1]->s=x->C[0]->s=x->s;
3058        x->tot=x->s*x->sz;
37f3        x->max=x->m1=x->mr=x->tot;
89b1        if (x->s<0)
be32          x->max=x->m1=x->mr=x->s;
95cf      }
e2e3      if (x->rev){
f24b        tmp=x->m1; x->m1=x->mr; x->mr=tmp;
ef88        ts=x->C[1]; x->C[1]=x->C[0]; x->C[0]=
          ts;
          x->C[1]->rev=!x->C[1]->rev; x->C
          [0]->rev=!x->C[0]->rev;
          }
          x->same=x->rev=0;
          }*/
          void route( SplayNode *k1, int c){
          SplayNode *k2=k1->f;
          //labledown(k2->C[!c]);labledown(k1->
          C[0]);labledown(k1->C[1]);
          k2->C[c]=k1->C[!c];
          k2->C[c]->f=k2;
          k1->f=k2->f;
          if (k2->f->C[0]==k2) k2->f->C[0]=k1;
          else k2->f->C[1]=k1;
          k2->f=k1;
          k1->C[!c]=k2;
          update(k2);//update(k1);
          if (root==k2) root=k1;
          }
          SplayNode * rank( int k){
          SplayNode *ts=root;
          int tmp;
          while (k){
          //labledown(ts);
          tmp=ts->C[0]->sz;
          if (k<=tmp) ts=ts->C[0];
          else if (k<=tmp+ts->c+1) break;
          else k-=tmp+ts->c+1, ts=ts->C[1];
          }
          return ts;
          }
          /*void select( int s, int r){
          rank(s,null);rank(r,root);
          }*/
          void splay( SplayNode *x, SplayNode *s)
          {
          if (x==null) return ;
          update(x);
          //labledown(x);
          while (x->f!=s){
          if (x->f->f==s){
          if (x->f->C[0]==x)
          route(x,0);
          else
          route(x,1);
          } else if (x->f->f->C[0]==x->f){
          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);
4f2d             return ;
95cf         }
35a1         else if (k<ts->s) ts=ts->C[0];
b87e         else ts=ts->C[1];
95cf     }
4efc     if (k<ls->s)
4506     {
7728         ls->C[0]=NewNode(ls,k);
96ed         update(ls->C[0]);update(ls);
4b8a         splay(ls->C[0],null);
d268     } else
4506     {
703f         ls->C[1]=NewNode(ls,k);
9628         update(ls->C[1]);update(ls);
758e         splay(ls->C[1],null);
95cf     }
95cf }
427e
427e //value k
6bd3 void remove( 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             {
c447                 if (ts->c==0)
4506                     {

```

```

splay(ts,null);
splay(rank(ts->C[0]->sz),root);
root=ts->C[0];
root->f=null;
root->C[1]=ts->C[1];
ts->C[1]->f=root;
update(root);
} else ts->c--,update(ts),splay(
ts,null);
return ;
}
else if (k<ts->s) ts=ts->C[0];
else ts=ts->C[1];
}
return ;//not find
}
//rank k
void del( int k)
{
splay(rank(k+1),null);
splay(rank(root->C[0]->sz),root);
SplayNode *ts=root;
root=ts->C[0];
root->f=null;
root->C[1]=ts->C[1];
ts->C[1]->f=root;
update(root);
}
int find( int k)
{
splay(rank(k+1),null);
return root->s;
}
int find_v( int k)
{
SplayNode *ts=root;
while (ts!=null){
//labledown(ts);
if (k==ts->s)
{
splay(ts,null);
return 1;
}
else if (k<ts->s) ts=ts->C[0];
else ts=ts->C[1];
}
return 0;
}
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[MAXN], lx[MAXN], rx[MAXN];
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     }

```

```

}node[MAXN];
int nodesize;
// 储存结果
struct CYL{
    POINT p;
    long long r;
    bool operator < (const CYL &a)const{
        return r < a.r;
    }
};
// 求出v 节点下的所有点到顶点p 的距离
void searchr(int v, const POINT &p,
priority_queue<CYL> &pq, int k){
    if (v == -1) return;
    CYL c;
    c.p = node[v].p;
    c.r = c.p.dis(p);
    pq.push(c);
    while (pq.size() > k) pq.pop();
    searchr(node[v].left, p, pq, k);
    searchr(node[v].right, p, pq, k);
}
// 贪心地返回点数接近k 的节点
int searchknode(int v, const POINT &p,
int k){
    if ((v == -1)||((node[v].number < k))
        return -1;
    cmpindex = node[v].degree;
    if (cmp(p, node[v].p)){
        if ((node[v].left != -1)&&(node[node[
v].left].number >= k))
            return searchknode(node[v].left, p,
                k);
    }else{
        if ((node[v].right != -1)&&(node[node[
v].right].number >= k))
            return searchknode(node[v].right, p
                , k);
    }
    return v;
}
// 检查某区域是否可能有k 小的点
bool check(POINT &root, const POINT &p,
priority_queue<CYL> &pq){
    POINT c;
    c.degree = p.degree;
    for (int i = 0; i < p.degree; ++i){
        if ((root.lx[i] <= p.x[i]&&(p.x[i]
            <= root.rx[i]))){
            c.x[i] = p.x[i];
            continue;
        }
        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){
    if (v == -1) return;
    if (node[v].left != -1){
        for (int i = 0; i < node[v].p.degree;
            ++i){
            node[node[v].left].p.rx[i] = node[v
                ].p.rx[i];
            node[node[v].left].p.lx[i] = node[v
                ].p.lx[i];
        }
        node[node[v].left].father = v;
        node[node[v].left].p.rx[node[v].
            degree] = node[v].p.x[node[v].
                degree];
        boundset(node[v].left);
    }
    if (node[v].right != -1){
        for (int i = 0; i < node[v].p.degree;
            ++i){
            node[node[v].right].p.rx[i] = node[
                v].p.rx[i];
            node[node[v].right].p.lx[i] = node[
                v].p.lx[i];
        }
        node[node[v].right].father = v;
        node[node[v].right].p.lx[node[v].
            degree] = node[v].p.x[node[v].
                degree];
        boundset(node[v].right);
    }
}
// 根据p 中的[s, t] 点建立Kd-TREE ,
//p 中的顺序会被破坏
//degree 是首先比较的维度
int buildtree(int s, int t, int degree,
    POINT p[], begin = 1){
    if (begin) nodesize = 0;
    if (s == t) return -1;
    else if(t - s == 1){
        node[nodesize].init(p[s], degree);
        return nodesize++;
    }
    cmpindex = degree;
    sort(p + s, p + t, cmp);
    int mid = (s + t - 1) >> 1;
    int v = nodesize;
    node[nodesize++].init(p[mid], degree);
    node[v].left = buildtree(s, mid, (
        degree + 1) % p[mid].degree, p, 0);
    node[v].right = buildtree(mid + 1, t, (
        degree + 1) % p[mid].degree, p, 0);
    node[v].number = t - s;
    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
427e //sa 中
427e
2278 int wa[maxn],wb[maxn],wv[maxn],ws_[maxn];
3aa5 int cmp(int *r,int a,int b,int l)
4506 {
e543     return r[a]==r[b]&&r[a+l]==r[b+l];
95cf }
427e //是数组的长度, 即 ( ) nrstrlenr,m 为语言集的
      最大编号
cdcb void da(int *r,int *sa,int n,int m)
4506 {
8fa8     int i,j,p,*x=wa,*y=wb,*t;
c934     for(i=0;i<m;i++) ws_[i]=0;
e6d3     for(i=0;i<n;i++) ws_[x[i]=r[i]]++;
2d9a     for(i=1;i<m;i++) ws_[i]+=ws_[i-1];
5b0a     for(i=n-1;i>=0;i--) sa[—ws_[x[i]]]=i;
427e
efbc     for(j=1,p=1;p<n;j*=2,m=p)
4506     {
5b7b         for(p=0,i=n-j;i<n;i++) y[p++]=i;
dc1c         for(i=0;i<n;i++) if(sa[i]>=j) y[p++] =
            sa[i]-j;
792f         for(i=0;i<n;i++) wv[i]=x[y[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];
e91d         for(i=n-1;i>=0;i--) sa[—ws_[wv[i]]]=
            y[i];

```

for(t=x,x=y,y=t,p=1,x[sa[0]]=0,i=1;i<n;i++)	0b94
x[sa[i]]=cmp(y,sa[i-1],sa[i],j)?p-1:p++;	fef3
}	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+1]==r[b+1];}	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
for(i=0;i<n;i++) ws[wv[i]]++;	7f78
for(i=1;i<m;i++) ws[i]+=ws[i-1];	be11
for(i=n-1;i>=0;i--) sa[-ws[wv[i]]]=y	f3a2
[i];	
for(t=x,x=y,y=t,p=1,x[sa[0]]=0,i=1;i<n;i++)	0b94
x[sa[i]]=cmp(y,sa[i-1],sa[i],j)?p-1:p	fef3
++;	
}	95cf
}/以下是	fe38
//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)	427e
*3+2)	
//int wa[maxn],wb[maxn],wv[maxn],ws_[maxn]	427e
];	
int c0(int *r,int a,int b)	9750
{	4506
return r[a]==r[b]&&r[a+1]==r[b+1]&&r[a	d7f5
+2]==r[b+2];	
}	95cf
int c12(int k,int *r,int a,int b)	6829
{ if(k==2) return r[a]<r[b] r[a]==r[b	1258
]&&c12(1,r,a+1,b+1);	
else return r[a]<r[b] r[a]==r[b]&&wv[8b9e
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
64aa     for(p=1,rn[F(wb[0])]=0,i=1;i<tbc;i++)
16b5     rn[F(wb[i])]=c0(r,wb[i-1],wb[i])?p-1:p
      ++;
427e
778a     if(p<tbc) dc3(rn,san,tbc,p);
6879     else for(i=0;i<tbc;i++) san[rn[i]]=i;
427e
37b9     for(i=0;i<tbc;i++) if(san[i]<tb) wb[ta
      ++]=san[i]*3;
427e
6cbc     if(n%3==1) wb[ta++]=n-1;
427e
0fde     sort_sa(r,wb,wa,ta,m);
427e
e658     for(i=0;i<tbc;i++) wv[wb[i]]=G(san[i])
      ]=i;
79d3     for(i=0,j=0,p=0;i<ta && j<tbc;p++)
427e
1a17     sa[p]=c12(wb[j]%3,r,wa[i],wb[j])?wa[i
      ++]:wb[j++];
427e
d05f     for(;i<ta;p++) sa[p]=wa[i++];
b151     for(;j<tbc;p++) sa[p]=wb[j++];
427e
95cf }
427e
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
      // ( ) 一个重要的结论:
      //suffix(j) 和suffix(k) 的最长公共前缀
      为height[rank[j]+1],
      //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
      //r 数组中, 从r到[0]r[n, 长度为-1], 且最大值
      小于
      //。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
      //sa 数组中, 从sa到[0]sa[n.-1]
      int rank[maxn],height[maxn];
      void calheight(int *r,int *sa,int n)
      {
      int i,j,k=0;
      for(i=1;i<=n;i++) rank[sa[i]]=i;
      for(i=0;i<n;height[rank[i++]]=k)
      for(k?k-:0,j=sa[rank[i]-1];r[i+k]==r[
      j+k];k++);
      return;
      }
      char str[100];
      int r[100];
      int sa[100];
      int main()
      {
      while(cin>>str){
      int L=strlen(str);
      for(int i=0;i<L;i++)r[i]=str[i]-'0';
      for(int i=0;i<L;i++)sa[i]=i;
      r[L]=0;
      da(r,sa,L+1,200);
      cout<<"haha\n";
      for(int i=0;i<L;i++)
      cout<<sa[i]<<' ';cout<<endl;
      calheight(r,sa,L);
      for(int i=0;i<L;i++)
      cout<<height[i]<<' ';cout<<endl;
      }
      }

```

5.5 线段树区间修改

```

#include <cstdio>
struct IT
{
#define MN 1048577*2
long long F[MN],D[MN];
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 }
25be void Add( 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     {
f1c1         D[k]+=C;
eed2         upd(k,l,r);
4f2d         return ;
95cf     }
817a     Add(k<<1,l,(l+r)>>1);
3afb     Add((k<<1)+1,((l+r)>>1)+1,r);
7514     F[k]=F[k<<1]+F[(k<<1)+1];
95cf }
4d16 void init( int NN)
4506 {
2944     for (N=1;N<NN;N<=1);
b3f6     for (int i=1;i<=NN;i++) scanf("%I64d
        ",&F[i+N-1]);
7991     for (int i=N-1;i;i--) F[i]=F[i<<1]+F
        [(i<<1)+1];
95cf }
a363 long long search( int _l, int _r)
4506 {
a0c0     ll=_l;rr=_r;
ea34     Ans=0;
cc62     find(1,1,N);
ba71     return Ans;
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
int N,M,a,b,c; 427e
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
#include<cstdio> 59b9
#include<cmath> c928
#include<vector> 09f7
#include<queue> acb9
#include<algorithm> 54ff
using namespace std; 421c
//const int inf=INT_MAX; 427e
//int maxInt = 0x7FFFFFFF;// 32 427e
bit
//long maxLong = 0x7FFFFFFFFFFFFFFF; 427e
// 64 bit
#define ll long long c115
const int inf= 0x7FFFFFFF; 5841
const char atcg[]="ATCG"; b43a
const int kind=4; 94ec
int n,m; 35b8
char in[105]; 586b
struct ahocorasick{ cbdf
    static const int undef=0;//初始化tag 5520
    static const int maxn=1<<10; df17
    static const int charset子节点个数=4;// 8c42
    int end; 4022
    int tag[maxn],fail[maxn],trie[maxn][ 2e34
        charset];
    void init(){ 5d53
        tag[0]=undef; 59e7
        fill(trie[0],trie[0]+charset,-1); 196d
        end=1; feb8
    } 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
6c43 trie[0][i]=0;
95cf }
88bb while(!bfs.empty()){
e42e int p=bfs.front();
38ff tag[p]|=tag[fail[p]];
1a76 bfs.pop();
6830 for(int i=0;i<charset;i++){
9f81 if(trie[p][i]!=-1){
076e fail[trie[p][i]]=trie[fail[p]][i];
        bfs.push(trie[p][i]);
95cf }
649a else
7720 trie[p][i]=trie[fail[p]][i];
95cf }
95cf }
0244 }ac;
7af7 bool dp[2][1<<11][1<<11];
1abe int sum[1<<11];
dfbc int w[101];
3117 int main(){
8b27 int i,j,k,t,cur,pre;
178e while(scanf("%d%d",&n,&m)+1){
c5bc ac.init();
ee09 for(i=0;i<n;i++){
137a scanf("%s %d",in,&w[i]);
aa86 ac.add(strlen(in),in,1<i);
95cf }

ac.build();
memset(sum,0,sizeof(sum));
for(i=0;i<(1<<10);i++)
for(j=0;j<n;j++){
if(i&(1<<j))
sum[i]+=w[j];
}
memset(dp,false,sizeof(dp));
cur=0,pre=1;
dp[cur][0][0]=true;
for(i=0;i<m;i++){
swap(cur,pre);
memset(dp[cur],false,sizeof(dp[cur]
));
for(int j=0;j<ac.end;j++)
for(k=0;k<4;k++){
int nxt=ac.trie[j][k];
for(t=0;t<(1<<n);t++)
dp[cur][nxt][t|ac.tag[nxt]] |=
dp[pre][j][t];
}
}
int ans=-1;
for(i=0;i<(1<<n);i++)
for(j=0;j<ac.end;j++){
if(dp[cur][j][i]){
ans=max(ans,sum[i]);
}
}
if(ans<0)puts("No Rabbit after
2012!");
else cout<<ans<<endl;
}
}

5.7 kmp

#include<iostream>
#include<cstdio>
#include<map>
#include<set>
#include<cstring>
#include<string>
using namespace std;

//extended kmp
//nxt[] & ext[] should be clarify out of
the function
void ExtendKmp(char s[],int ls,char t[],
int lt)
{
int i,j,k;
int Len,L;

```

```

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

```

5.8 Palindromic

```

cce1 void manacher( int* r, int len, int* p )
4506 {
cc69     int i, id, right = 0;

```

```

for( i = 0; i < len; ++i )
{
    if( i < right )
        p[i] = min(p[2*id-i], right-i);
    else
        p[i] = 1;
    while( i+p[i] < len && i-p[i] >= 0
        && r[i+p[i]] == r[i-p[i]] )
        p[i]++;
    if( p[i] + i > right )
    {
        right = p[i]+i;
        id = i;
    }
}
}

```

6 图论

6.1 Hamilton

```

//Hamilton 回路存在条件:
// 每个点的度数超过一半点数
// 复杂度 N^3 左右
//@
#include <stdio>
#include <cstring>

#define MN 201

int map[MN][MN];
int N,M;
int ans[MN];

void reverse( int ans[MN], int s, int t )
{
    int temp;
    while (s<t)
    {
        temp=ans[s];
        ans[s]=ans[t];
        ans[t]=temp;
        s++;t--;
    }
}

void Hamilton()
{
    int s=1,t;
    int ansi=2,i,j,w,temp;
    bool vis[MN]={false};
    memset(vis,0,sizeof vis);
    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;
1f75         while (1)
4506         {
7790             for (i=1;i<=N;i++)
44dc                 if (map[t][i]&&!vis[i])
4506                 {
5260                     ans[ansi++]=i;vis[i]=1;
5093                     t=i;
6173                     break;
95cf                 }
882c             if (i>N) break;
95cf         }
fd5b         if (!map[s][t])
4506         {
110b             for (i=1;i<ansi-2;i++)
2fc4                 if (map[ans[i]][t]&&map[s][ans[i]
+1])) break;
4fa4             w=ansi-1;
a42b             i++;
642b             t=ans[i];
8ed9             reverse(ans,i,w);
95cf         }
34fc         if (ansi==N) return;
5cf5         for (j=1;j<=N;j++)
4506         {
ccf8             if (vis[j]) continue;
110b             for (i=1;i<ansi-2;i++)
2d6a                 if (map[ans[i]][j]) break;
2d6a                 if (map[ans[i]][j]) break;
95cf         }
7dd0         s=ans[i-1];

```

```

t=j;
reverse(ans,0,i-1);
reverse(ans,i,ansi-1);
ans[ansi++]=j;
vis[j]=1;
}
}
int main()
{
while (scanf("%d%d",&N,&M)!=EOF)
{
memset(map,0,sizeof map);
int a,b;
while (M--)
{
scanf("%d%d",&a,&b);
map[a][b]=map[b][a]=1;
}
Hamilton();
for (int i=0;i<N-1;i++) printf("%d ",
ans[i]);
printf("%d\n",ans[N-1]);
for (int i=0;i<N-1;i++)
if (!map[ans[i]][ans[(i+1)%N]])
while (1);
}
return 0;
}

```

6.2 HopcraftKarp

```

// HK 解二分图匹配
// 复杂度 O(sqrt(N)*M)
/*@
#include <stdio>
#include <cstring>

#define MN 50001
#define MM 150001

struct E_Node
{
int v,ne;
} E[MM];

int R[MN],e_sz;
int N,M;

void Add( int a, int b)
{
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             }
4506             {
19cf                 Qu[++qsz]=Cy[v];
a158                 Lx[Cy[v]]=Ly[v]+1;
95cf             }
95cf         }
95cf     }
95cf     return flag;
01d8 }
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;

```

```

Cx[k]=E[p].v;
return 1;
}
}
return 0;
}

Add(a,b);

ans=0;
memset(Cx,0xff,(N+1)*sizeof(Cx[0]));
memset(Cy,0xff,(M+1)*(sizeof(Cy[0])))
;
while (BFS())
{
    for (int i=N;i;i--)
        if (Cx[i]==-1&&Dfs(i)) ans++;
}
printf("%d\n",ans);
}

```

6.3 HopcraftKarp

```

// HK 解二分图匹配复杂度
// O(sqrt(N)*M)
/*@
#include <stdio>
#include <cstring>

#define MN 50001
#define MM 150001

struct E_Node
{
    int v,ne;
} E[MM];

int R[MN],e_sz;
int N,M;

void Add( int a, int b)
{
    E[++e_sz].v=b;E[e_sz].ne=R[a];R[a]=e_sz
    ;
}

void init()
{
    e_sz=0;
    memset(R,0,sizeof R);
}

```

```

427e int ans, qsz;
8a67
427e int Qu[MN*2], Lx[MN], Ly[MN], Cx[MN], Cy[MN];
8ec5
427e bool BFS()
e459
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             }
4506             {
19cf                 Qu[++qsz]=Cy[v];
a158                 Lx[Cy[v]]=Ly[v]+1;
95cf             }
95cf         }
95cf     }
95cf     return flag;
01d8 }
427e bool Dfs( int k)
ec04
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 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-->0)
3c73        if (Cx[i]==-1 && Dfs(i)) ans++;
3ed3
}
95cf printf("%d\n", ans);
53b1
}
95cf

```

6.4 Hungary

```

// 匈牙利算法解二分图匹配
// 一般小于 NM
// *@
427e bool Vis[MN];
e219 int Link[MN];
bf70 int Cache[MN];
2a22 int Csz;
2fd8
427e bool find( int k)
b96a
{
4506
    for (int p=R[k]; p; p=E[p].ne)
0470        if (!Vis[E[p].v])
4d33        {
4506            Vis[E[p].v]=1;
39cb            Cache[++Csz]=E[p].v;
6bbc            if (!Link[E[p].v])
f267            {
4506                Link[E[p].v]=k;
17ae                return true;
3361            }
95cf            if (find(Link[E[p].v]))
fc61            {
4506                Link[E[p].v]=k;
17ae                return true;
3361            }
95cf        }
95cf    }
438e    return false;
95cf
}
427e
memset(Link, 0, sizeof Link);
f531
for (int i=1; i<=N; i++)
c802
{
4506
    for (int k=1; k<=Csz; k++)
73c9        Vis[Cache[k]]=0;
c926        Csz=0;
19e8        if (find(cc(i, j))) ans++;
8f54
}
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 <stdio>
ef2f #include <string>
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     {
aab6         if(visy[y])
b333             continue;
44b7         int t = lx[x] + ly[y] - w[x][y];
d790         if(t==0)
4506         {
d2aa             visy[y] = true;
6def             if(linky[y]==-1 || find(linky[y]))
4506             {
b930                 linky[y] = x;
3361                 return true;
95cf             }
95cf         }
cf8b         else if(slack[y] > t)
52e5             slack[y] = t;
95cf     }
438e     return false;
95cf }
427e
45ef int KM()
4506 {
576f     int i,j;
427e
76e3     memset(linky,-1,sizeof(linky));
b7b2     memset(ly,0,sizeof(ly));
7790     for(i = 1; i <=N; i++)
4506     {
7b3c         lx[i] = -INF;
5cf5         for(j = 1; j <=N; j++)
7f0c             if(w[i][j] > lx[i])
c66c                 lx[i] = w[i][j];

```

```

    }
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++)
7790
    {
4506
        if(visy[i])
4427
            ly[i] += d;
ab77
        else
649a
            slack[i] -= d;
a397
    }
95cf
}
95cf
}
95cf
int result = 0;
bf72
for(i = 1; i <=N; i++)
7790
if(linky[i]>-1)
f7d9
    result += w[linky[i]][i];
a453
return result;
56b0
}
95cf
427e
int main()
299c
{
4506
    while (scanf("%d",&N)!=EOF)
3dda
    {
4506
        memset(w,0,sizeof w);
39be
        for (int i=1;i<=N;i++)
c802
            for (int j=1;j<=N;j++) scanf("%d",&
f595
                w[i][j]);
05c2
        printf("%d\n",KM());
95cf
    }
95cf
}

```

6.6 lca

```
// 建树和倍增求 lca
```

```
427e
```



```

427e //buildtree
427e
427e
727b int dep[MN],root[MN];
0ae7 int ancestor[MN][logMN];
79ea void buildtree( int k)
4506 {
0b49     vis[k]=1;
0470     for (int p=R[k];p;p=E[p].ne)
4506     {
f90d         if (vis[E[p].v]) continue;
b082         ancestor[E[p].v][0]=k,dep[E[p].v]=
            dep[k]+1,root[E[p].v]=root[k],
            buildtree(E[p].v);
95cf     }
95cf }
427e //lca_init
427e // 可以处理森林
1b95 void lca_init( int N)
4506 {
5750     memset(ancestor,0,sizeof ancestor);
91c5     for (int i=1;i<=N;i++) vis[i]=0;
c802     for (int i=1;i<=N;i++)
014c         if (ancestor[i][0]==0) dep[i]=1,
            root[i]=i,buildtree(i);
427e
4a2e     for (int k=1;k<=log2(N);k++)
c802         for (int i=1;i<=N;i++)
98fc             ancestor[i][k]=ancestor[ancestor[
                i][k-1]][k-1];
95cf }
427e
427e //return lca(a,b)
427e //return -1 if a,b in diffirent tree
82b0 int lca_query( int a, int b, int N)
4506 {
bb4f     int ans=dep[a]+dep[b];
5801     if (root[a]!=root[b]) return -1;
686e     if (a==b) return a;
6371     if (dep[a]>dep[b]) swap(a,b);
798a     for (int k=0;k<=log2(N);k++) if ((1<<
        k)&delta) b=ancestor[b][k];
686e     if (a==b) return a;
7b0a     for (int k=log2(N);k>=0;k--) if (
        ancestor[a][k]!=ancestor[b][k]) a=
        ancestor[a][k],b=ancestor[b][k];
b01a     return ancestor[a][0];
95cf }

```

6.7 MaxcostMaxflow

```

427e // 最小费用路增广 bySPFA
427e //*@
59b9 #include <cstdio>

```

```

#include <cstring> ef2f
#include <queue> acb9
427e
using namespace std; 421c
427e
#define MN 30 e8b6
#define MM 100000 8c3b
#define INF 0x7fffffff 1cc6
427e
int T,N,M,K; aa81
int Lim[MN],Like[MN][MN],l,cases; cb35
427e
struct MaxcostMaxflow 0c77
{ 4506
    struct E_Node 673f
    { 4506
        int v,f,c,op,ne; 8253
    } E[MM]; f9e1
    int R[MN],sz; 5586
    int S,T; 9bfc
    void Add( int a, int b, int c, int d) cb15
    { 4506
        E[++sz].v=b;E[sz].ne=R[a];R[a]=sz;E[ 233c
            sz].f=c;E[sz].c=d;
95cf    }
void Ins( int a, int b, int c, int d) d3c6
{ 4506
    if (!c) return ; 4238
    Add(a,b,c,d);E[sz].op=sz+1; 18b4
    Add(b,a,0,-d);E[sz].op=sz-1; 11e8
} 95cf
bool B[MN]; 7d24
int D[MN],F[MN]; 0e46
int maxcost,maxflow; 3d1d
void init() 88f1
{ 4506
    sz=0; 1bb9
    memset(R,0,sizeof R); ab17
    maxcost=maxflow=0; 1fd4
} 95cf
queue<int> Q; aaaf
bool SPFA() 64de
{ 4506
    int u,m,p; 3ccf
    while (!Q.empty()) Q.pop(); a058
    Q.push(S); e753
    for (int i=1;i<=T;i++) D[i]=-INF,B[i] 5480
        ]=0;
    D[S]=0;B[S]=1; eb5f
    while (!Q.empty()) 1b18
    { 4506
        u=Q.front(); 50ae
        Q.pop(); f2f8
    }
}

```

```

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

6.8 NetworkFlow

427e     //Dinic
// 当前弧、多路增广、断层优化
/*@
#include <cstring>
#define MN 1000
#define MM 100000

struct E_Node
{
    int a,b,opp,ne;
    int f;
} E[MM];

int R[MN],e_sz;

struct s_node{
    int v,p;
} s[MN];

int top,N,M,S,T,ans;
int Q[MN],L[MN];
bool hash[MN];

void init()
{
    memset(R,0,sizeof R);
    e_sz=0;
    ans=0;
}

void add( int a, int b, int c){
    E[++e_sz].a=a;E[e_sz].b=b;E[e_sz].f=c;
    E[e_sz].ne=R[a];R[a]=e_sz;
}

void ins( int a, int b, int c){
    add(a,b,c);
    E[e_sz].opp=e_sz+1;
    add(b,a,0);
    E[e_sz].opp=e_sz-1;
}

int BFS(){
    int h,t,p;
    memset(L,0,sizeof(L));
    Q[h=t=1]=S;
    L[S]=1;
    for (;h<=t;h++){
        for (p=R[Q[h]];p;p=E[p].ne)
            if (E[p].f&&(!L[E[p].b])){
                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
9690 void Dinic(){
2f55     bool f;
a591     int *p,*v;
10a0     s[top=1].v=S;s[1].p=0;
d9aa     memset(hash,true,sizeof(hash));
55e4     while (top){
0cc3         v=&s[top].v;
ac2c         p=&s[top].p;
3605         if (*v==T){
454a             top=augmented();
b333             continue;
95cf         }
607d         if (*p)
c70d             *p=E[*p].ne;
649a         else
27e8             *p=R[*v];
3b7e         f=false;
9933         for (;*p;*p=E[*p].ne)
4eeb             if (hash[E[*p].b]&&E[*p].f&&L[*v]
b97a                 +1==L[E[*p].b]){
5fad                 s[++top].v=E[*p].b;
6aed                 s[top].p=0;
69dc                 top++;
6173                 f=true;
95cf                 break;
d954             }
cdee             if (!f)
e9a7                 hash[s[top].v]=false;
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
    struct E_Node{
a981
        int v,ne;
28a5
    } E[MM];
f9e1
    int R[MN],sz;
5586
    void Add( int a, int b){
0d5d
        E[++sz].v=b;E[sz].ne=R[a];R[a]=sz;
1352
    }
95cf
    stack<int> S;
870e
    bool B[MN];
7d24
    int D[MN],L[MN],C[MN],SZ[MN];
b156
    int nowc,nowd;
85a3
    void Dfs( int k){
9372
        D[k]=L[k]=++nowd;
0c3d
        B[k]=1;S.push(k);
d651
        for (int p=R[k];p;p=E[p].ne)
0470
            if (B[E[p].v]) L[k]=min(L[k],D[E[p]
2f4c
                ].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 <stdio>
ef2f    #include <string>
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    int R[MN],e_sz;
7c0e
427e
427e    void Add( int a, int b, int c=0)
c5de    {
4506        if (a==b) return;
e53f        E[++e_sz].v=b;;E[e_sz].ne=R[a];R[a]=
98b5        e_sz;E[e_sz].a=a;
427e    }
95cf
427e
427e    int SZ[MN];
4c87
80ca    int O[MM];
427e
88f1    void init()
4506    {
1282        e_sz=0;
ab17        memset(R,0,sizeof R);
427e
427e        //Edge_Connect
427e    }
95cf
427e
427e    //Rebuild重构图
427e    //
427e    //E[],R[],N,col[],color
427e    //G
427e    int _col[MN];
060e
427e
427e    void Rebuild( Graph &GG)
6e8b    {
4506        GG.init();
2776        for (int i=1;i<=color;i++) GG.cut[i]
f4f9            =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 //
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=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;

else if (dfs[k]!=1&&dfs[k]<=low[E[p
].v]) cut[k]=1;
if (dfs[k]<=low[E[p].v])
{
    cedge.clear();
    col[k]=++color;
    col[E[p].v]=color;
    cedge.pb(k);
    cedge.pb(E[p].v);
    while (stk.top()!=E[p].v)
    {
        col[stk.top()]=color;
        cedge.pb(stk.top());
        stk.pop();
    }
    stk.pop();
    Cedge();

    //stk.pop();
}
}
#undef p

void Edge_Connect()
{
    memset(cut,0,sizeof cut);
    while (stk.size()) stk.pop();
    for (int i=0;i<=e_sz;i++) vise[i]=0;

    color=cc=0;
    for (int i=1;i<=N;i++)
        vis[i]=0,col[i]=0;
    for (int i=1;i<=N;i++)
    {
        cc=0;
        if (!vis[i]) Dfs(i);
        if (stk.size())
        {
            ++color;
            while (stk.size())
            {
                col[stk.top()]=color;
                stk.pop();
            }
        }
    }
}

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

int cases;

int main()
{
    int a,b;
    while (scanf("%d%d",&N,&M)!=EOF)
    {
        G.init();
        //Edge.clear();
        for (int i=1;i<=M;i++)
        {
            scanf("%d%d",&a,&b);
            a++;b++;
            // if (Edge.find(make_pair(a,b))!=
            Edge.end()) continue;
            G.Add(a,b);G.Add(b,a);
            // Edge.insert(make_pair(a,b));
            // Edge.insert(make_pair(b,a));
        }
        G.Edge_Connect();
        G.Rebuild(GG);
        GG.lca_init(G.color);

        int T;
        scanf("%d",&T);
        printf("Case #%d:\n",++cases);
        while (T--)
        {

```

```

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 {
576f     int i,j;
2dbf     for (i=0;i<n;i++)
0180         if (i!=x) map[x][i]+=map[y][i],map[i]
          ][x]+=map[i][y];
ffcb     for (i=y+1;i<n;i++) for (j=0;j<n;j++)
4506         {
ab88             map[i-1][j]=map[i][j];
c988             map[j][i-1]=map[j][i];
95cf         }
61b6         n--;
95cf     }
78af     int w[maxn],c[maxn];
d7dd     int sx,tx;
9acc     int mincut()
4506     {
74c3         int i,j,k,t;
7dd1         memset(c,0,sizeof(c));
3bab         c[0]=1;
457c         for (i=0;i<n;i++) w[i]=map[0][i];
a17f         for (i=1;i+1<n;i++)
4506         {
901d             t=k=-1;
4d7b             for (j=0;j<n;j++) if (c[j]==0&&w[j]>k
                )
8647                 k=w[t=j];
d40b             c[sx=t]=1;
e5e3             for (j=0;j<n;j++) w[j]+=map[t][j];
95cf         }
6bff         for (i=0;i<n;i++) if (c[i]==0) return w
            [tx=i];

```

```

    }
int main()
{
    int i,j,k,m;
    while (scanf("%d%d",&n,&m)!=EOF)
    {
        memset(map,0,sizeof(map));
        while (m--)
        {
            scanf("%d%d%d",&i,&j,&k);
            map[i][j]+=k;
            map[j][i]+=k;
        }
        int mint=999999999;
        while (n>1)
        {
            k=mincut();
            if (k<mint) mint=k;
            contract(sx,tx);
        }
        printf("%d\n",mint);
    }
    return 0;
}

```

6.12 度限制生成树

```

#include <iostream>
#include <fstream>
#include <climits>
#include <queue>
#include <map>
#include <cstring>
#include <string>
using namespace std;
const int maxn = 25;
struct node
{
    int v, w;
};
struct cmp
{
    bool operator() (const node &a, const
        node &b)
    {
        return a.w > b.w; // 是从小到大>.
    }
};
int n, m, s; // n 个点, m 条边, s 为原
点. 点为有度数限制的点V0
int num;
int minV0[maxn];
int total; // 限制的度数.
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;
1a39         pre[i] = -1;
5037         for (j = 0; j <= maxn - 1; j++)
4506         {
b4df             g[i][j] = INT_MAX;
95cf         }
95cf     }
91c9     n = 0;
2eb3     cin >> m;
6988     for (i = 1; i <= m; i++)
4506     {
fad5         cin >> name1 >> name2 >> w;
a3dd         iter = Map.find(name1);
917d         if (iter == Map.end())
4506         { // 说明该结点还不存在。
c93c             n++;
8edc             Map[name1] = n;
95cf         }
fbab         a = Map[name1];
a84b         iter = Map.find(name2);
917d         if (iter == Map.end())
4506         { // 说明该结点还不存在。
c93c             n++;
e2a7             Map[name2] = n;
95cf         }
3878         b = Map[name2];
a996         if (g[a][b] > w)
4506         {
ff0f             g[a][b] = g[b][a] = w;
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
    int i, j, k; c8ed
    node mini, temp; 3495
    while (!Q.empty()) Q.pop(); a058
    dist[s] = 0; c7c1
    temp.v = s; temp.w = 0; a8ed
    Q.push(temp); 6b09
    for (k = 1; k <= n; k++) e0e5
    { 4506
        while (!Q.empty()) 1b18
        { 4506
            mini = Q.top(); c3cc
            Q.pop(); f2f8
            j = mini.v; 6e31
            if (!p[j]) 6dbf
            { 4506
                p[j] = 1; cde5
                if (g[0][j] < g[0][minV0[num]]) 37a1
                { 4506
                    minV0[num] = j; 0423
                } 95cf
                for (i = 1; i <= n; i++) 1f5c
                { 4506
                    if (i != j && !p[i] && dist[i] ae84
                        > g[j][i])
                    { 4506
                        dist[i] = g[j][i]; cf01
                        pre[i] = j; 671b
                        temp.w = dist[i]; temp.v = i; d413
                        Q.push(temp); 6b09
                    } 95cf
                }
            }
        }
    }
}

```



```

95cf        }
6173        break;
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
990e     for (i = top - 1; i >= 0; i--)
4506     {
5cac         j = Stack[i]; k = Stack[i + 1];
74d2         if (max_value[k] > g[j][pre[j]])
4506         {
30b8             max_value[j] = max_value[k];
4b3f             max_value_v[j] = max_value_v[k];
95cf         }
649a     else
4506     {
abf1         max_value[j] = g[j][pre[j]];
8b10         max_value_v[j] = j;
95cf     }
95cf }
95cf }
a038 void Solve(void)
4506 {
80a8     int i, j, k, l;
9059     int mini, opti_i, opti_maxV;
dc9d
0e77     for (k = 1; k <= num; k++)
4506     {
9bcb         ans += g[0][minV0[k]];
427e
8d9b         j = minV0[k]; i = pre[j];
c32d         while (i != -1)
4506         {
ae09             l = i;
00ff             i = pre[l];

```

```

        pre[l] = j;
        j = l;
    }
    pre[minV0[k]] = 0;
}
memset(p, 0, sizeof(p));
for (i = 1; i <= n; i++)
{
    if (!p[i])
    {
        Cal_max_value(i);
    }
}
for (k = 1; k <= total - num; k++)
{
    mini = 0;
    for (i = 1; i <= n; i++)
    {
        if (pre[i] == 0) continue;
        if (g[0][i] - max_value[i] < mini)
        {
            mini = g[0][i] - max_value[i];
            opti_i = i; opti_maxV = max_value_v
                [i];
        }
    }
    if (mini == 0) break;
    ans += mini;
    pre[opti_maxV] = -1;
    j = opti_i; i = pre[j];
    while (i != -1)
    {
        l = i;
        i = pre[l];
        pre[l] = j;
        j = l;
    }
    pre[opti_i] = 0;
    Cal_max_value(opti_maxV);
}
}

```

6.13 最小树形图

```

#include <iostream>
#include <cstdio>
#include <cstring>
#include <cmath>
#define MAXN 128
#define MAXM 32768
#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 }
28f2 bool init()
4506 {
f547     if (scanf("%d%d", &n, &m) == EOF)
        return false;
eec6     double x[MAXN], y[MAXN];
e635     int a, b;
2ad4     for (int i = 1; i <= n; ++i)
3c14         scanf("%lf%lf", &x[i], &y[i]);
2ad4     for (int i = 1; i <= n; ++i)
8c5b         for (int j = 1; j <= n; ++j)
f5a0             g[i][j] = INF;
2af5     for (int i = 1; i <= m; ++i)
4506     {
a6b8         scanf("%d%d", &a, &b);
3126         g[a][b] = min(g[a][b], dist(x[a], y[a]
            ], x[b], y[b]));
95cf     }
427e     //print_map();
3361     return true;
95cf }
c753 bool vst[MAXN];
8aca void dfs(int v)
4506 {
84ad     vst[v] = true;
2ad4     for (int i = 1; i <= n; ++i)
7532         if (!vst[i] && g[v][i] != INF) dfs(i)
        ;
95cf }
1a66 bool possible(int v)
4506 {

```

```

memset(vst, false, sizeof(vst));
dfs(v);
for (int i = 1; i <= n; ++i)
    if (i != v && !vst[i]) return false;
return true;
}
int pre[MAXN];
bool del[MAXN];
void solve(int v)// 根为v
{
    res = 0;
    int num = n;
    memset(del, false, sizeof(del));
    while(1)
    {
        int i;
        // 更新数组pre
        for (i = 1; i <= n; ++i)
        {
            if (del[i] || i == v) continue;
            pre[i] = i;
            g[i][i] = INF;
            for (int j = 1; j <= n; ++j)
            {
                if (del[j]) continue;
                if (g[j][i] < g[pre[i]][i])
                    pre[i] = j;
            }
        }
        for (i = 1; i <= n; ++i)
        {
            // 找环
            if (del[i] || i == v) continue;
            int j = i;
            memset(vst, 0, sizeof(vst));
            while (!vst[j] && j != v)
            {
                vst[j] = true;
                j = pre[j];
            }
            if (j == v) continue;
            i = j;
            // 更新, 有向环缩点res
            res += g[pre[i]][i];
            for(j = pre[i]; j != i; j = pre[j])
            {
                res += g[pre[j]][j];
                del[j] = true;
            }
            for(j = 1; j <= n; ++j)
            {
                if(del[j]) continue;
                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
6bff if(i > n){
2ad4     for(int i = 1; i <= n; ++i)
4506     {
50c1         if(del[i] || i == v) continue;
b0c8         res += g[pre[i]][i];
95cf     }
6173     break;
95cf }
95cf }
95cf }
299c int main()
4506 {
1f56     while (init())
4506     {
a248         if (!possible(1)) printf("poor snoopy
            \n");
649a         else
4506         {
1d60             solve(1);
d6bf             printf("%.2f\n", res);
95cf         }
95cf     }
7021     return 0;
95cf }

```

6.14 多重匹配

```

1915 #include <stdio.h>
0cbb const int maxn=100;
a23c int a[maxn],b[maxn],nov_a[maxn],nov_b[

```

```

maxn];
int w[maxn][maxn];
int m,n;
bool find(int i)
{
    nov_a[i]=0;
    for(int p=0;p<n;p++)
    if(w[i][p]==0&&nov_b[p])
    {
        nov_b[p]=0;
        if(b[p]>0)
        {
            b[p]--;
            w[i][p]=1;
            return true;
        }
    }
    for(int q=0;q<m;q++)
    if(w[q][p]==1&&nov_a[q])
    {
        if(find(q))
        {
            w[i][p]=1;
            w[q][p]=0;
            return true;
        }
    }
    return false;
}
bool gao(int i)
{
    for(int j=0;j<m;j++)nov_a[j]=1;
    for(int j=0;j<n;j++)nov_b[j]=1;
    return find(i);
}
int main()
{
    int cas=0;
    while(scanf("%d%d",&m,&n)!=EOF&&(m|n))
    {
        for(int i=0;i<m;i++)scanf("%d",&a[i])
;
        for(int i=0;i<n;i++)scanf("%d",&b[i])
;
        for(int i=0;i<m;i++)
            for(int j=0;j<n;j++)
                w[i][j]=0;
        bool ans=true;
        for(int i=0;i<m;i++)
        {
            while(a[i]>0&&gao(i))a[i]--;
            if(a[i]!=0)
            {

```

```

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

```