Algorithm

n+e F

2018 年 11 月 6 日

目录

1 FastIn+FastGetchar 1

2 FastOut+FastPutchar 1

3 数据结构 2
- 3.1 并查集 . . . 2
- 3.2 二叉堆 . . . 2
- 3.3 Hash . . . 2
- 3.4 树状数组 . . . 3
- 3.5 ZKW Segtree . . . 3
- 3.6 斜堆 . . . 3
- 3.7 可持久化 Treap . . . 3
- 3.8 Splay . . . 4
- 3.9 主席树 . . . 4
- 3.10 树链剖分 . . . 4
- 3.11 LCT . . . 5
- 3.12 KD-Tree . . . 6

4 DP 7
- 4.1 斜率优化 . . . 7

5 图论 7
- 5.1 边表 . . . 7
- 5.2 spfa . . . 7
- 5.3 Dijkstra . . . 8
- 5.4 二分图匹配-匈牙利 . . . 8
- 5.5 Dinic . . . 8
- 5.6 MCMF . . . 9
- 5.7 LCA-倍增 . . . 10
- 5.8 LCA-树链剖分版 . . . 10
- 5.9 带花树 . . . 10

6 字符串 11
6.1 Trie 11
6.2 KMP 11
6.3 AC 自动机 11
6.4 回文自动机 12
6.5 SA 带 height 12
6.6 SAM 13

7 数学 13
7.1 快速幂 13
7.2 exgcd 13
7.3 线性求逆元 13
7.4 线性筛求 φ 13
7.5 Miller-Rabin + Rho 14
7.6 高斯消元 15
7.7 矩阵快速幂 15
7.8 k 阶线性递推-特征多项式 16
7.9 拉格朗日插值 16
7.10 牛顿插值 17
7.11 BSGS 17
7.12 阶、原根 18
7.13 exBSGS 18
7.14 $O(n^{2/3})$ 求 φ/μ 的前缀和 18
7.15 单纯形 19
7.16 FFT 20
7.17 NTT 20
7.18 多项式求逆 21
7.19 找规律 22

8 计算几何 23
8.1 Point 的定义 (在复数域上) 23
8.2 二维凸包 23
8.3 三维凸包 23
8.4 半个半平面交 24
8.5 矩形面积并 25
8.6 曼哈顿距离最小生成树 26
8.7 欧几里德距离最小生成树 28
8.8 平面图点定位 30
8.9 辛普森积分 33
8.10 面积并-格林公式 35

1 FastIn+FastGetchar

```
char ch,B[1<<15],*S=B,*T=B;
#define getc() (S==T&&(T=(S=B)+fread(B,1,1<<15,stdin),S==T)?0:*S++)
#define isd(c) (c>='0'&&c<='9')
int aa,bb;int F(){
    while(ch=getc(),!isd(ch)&&ch!='-');ch=='-'?aa=bb=0:(aa=ch-'0',bb=1);
    while(ch=getc(),isd(ch))aa=aa*10+ch-'0';return bb?aa:-aa;
}
int F(){
    while(ch=getc(),ch<'-');int aa=ch-'0';
    while(ch=getc(),ch>='0')aa=aa*10+ch-'0';return aa;
}
```

2 FastOut+FastPutchar

```
char U[1<<15|1],*O=U,*W=U+(1<<15),stk[20],ts;
#define mod 1000000000
#define putc(c) (O==W?fwrite(U,1,1<<15,stdout),O=U,1:1,*O++=c)
#define pr(x) (x>=mod?pri(x/mod),pr9(x%mod):pri(x))
void pri(int x){
    if(!x)putc('0');else{
        for(ts=0;x;x/=10)stk[++ts]=x%10+'0';
        for(;ts;putc(stk[ts--]));
    }
}
void pr9(int x){
    for(ts=1;ts<=9;ts++)stk[ts]=x%10+'0',x/=10;
    for(ts=9;ts;putc(stk[ts--]));
}
```

3 数据结构

3.1 并查集

```
int fa[N];
int gf(int x){return fa[x]==x?x:fa[x]=gf(fa[x]);}
```

3.2 二叉堆

```
int hr;struct H{int n,d;}h[M];
void ins(const H&t){
    int i;
    for(i=++hr;i!=1&&t.d<h[i>>1].d;i>>=1)h[i]=h[i>>1];
    h[i]=t;
```

```
}
void dlt(){
    hr--;int i;
    for(i=1;(i<<1)<=hr&&!(h[hr+1].d<=h[i<<1].d&&h[hr+1].d<=h[i<<1|1].d);)
    i<<1!=hr&&h[i<<1|1].d<h[i<<1].d?(h[i]=h[i<<1|1],i=i<<1|1):(h[i]=h[i<<1],
        i=i<<1);
    h[i]=h[hr+1];
}
```

3.3 Hash

```
const int M=(1<<20)-1;
class HASH{public:
    int la[1<<20],et;
    struct E{long long key;int f,nxt;}e[N];
    void clr(){et=0;}
    inline void add(long long x,int f){
        for(int i=la[x&M];i;i=e[i].nxt)
        if(e[i].key==x){
            e[i].f=f;return;
        }
        e[++et]=(E){x,f,la[x&M]},la[x&M]=et;
    }
    inline void del(long long x){la[x&M]=0;}
    inline int query(long long x){
        for(int i=la[x&M];i;i=e[i].nxt)
        if(e[i].key==x)return e[i].f;
        return -1;
    }
}hash;
```

3.4 树状数组

```
int n,z[N];
void add(int t,int x){for(;t<=n;t+=t&-t)z[t]+=x;}
int getsum(int t){int f=0;for(;t;t-=t&-t)f+=z[t];return f;}
```

3.5 ZKW Segtree

```
for(n=F(),m=F(),d=1;d<n;d<<=1);
for(i=0;i<n;i++)t[i+d]=F();
for(i=d-1;i;i--)t[i]=t[i<<1]>t[i<<1|1]?t[i<<1]:t[i<<1|1];
while(m--)
if(F())for(t[i=d+F()-1]=F();i>>=1;t[i]=t[i<<1]>t[i<<1|1]?t[i<<1]:t[i<<1|1]);
else{
    for(l=F()+d-2,r=F()+d,o=0;l^r^1;l>>=1,r>>=1)
        ~l&1&&o<t[l^1]?o=t[l^1]:1,
```

```
        r&1&&o<t[r^1]?o=t[r^1]:1;
    printf("%d\n",o);
}
```

3.6 斜堆

```
int merge(int x,int y){
    if(!x||!y)return x|y;
    if(v[x]<v[y])swap(x,y);
    R[x]=merge(R[x],y);swap(L[x],R[x]);
    return x;
}
```

3.7 可持久化 Treap

```
struct T{int l,r,siz;char ch;}t[1<<23];
#define ls t[o].l
#define rs t[o].r
int merge(int l,int r){
    if(!l||!r)return l+r;int o=++tot;
    if(rand()%(t[l].siz+t[r].siz)<t[l].siz)
    t[o]=t[l],rs=merge(rs,r);
    else t[o]=t[r],ls=merge(l,ls);
    t[o].siz=t[l].siz+t[r].siz;
    return o;
}
void split(int x,int k,int&l,int&r){
    if(!x){l=r=0;return;}int o=++tot;t[o]=t[x];
    if(t[ls].siz>=k)split(ls,k,l,r),ls=r,r=o;
    else split(rs,k-1-t[ls].siz,l,r),rs=l,l=o;
    t[o].siz=t[ls].siz+t[rs].siz+1;
}
```

3.8 Splay

```
int fa[N],son[N],rev[N],siz[N];
#define ls son[o][0]
#define rs son[o][1]
void pu(int t){...}//push_up
void pd(int t){...}//push_down
void rtt(int t){//rotate
    int f=fa[t],p=son[f][1]==t;
    (fa[t]=fa[f])?son[fa[f]][son[fa[f]][1]==f]=t:1;
    (son[f][p]=son[t][!p])?fa[son[f][p]]=f:1,
    pu(son[fa[f]=t][!p]=f);
}
void pv(int t,int r){//preview
```

```
    if(fa[t]!=r)pv(fa[t]);pd(t);
}
void splay(int t,int root=0){int f;
    for(pv(t,root);fa[t]!=root;rtt(t))
    if(f=fa[t],fa[f]!=root)rtt(son[f][1]==t^son[fa[f]][1]==f?t:f);//delete?
    pu(t);
}
int pre(int o){for(splay(o),o=ls;rs;o=rs);return o;}
int nxt(int o){for(splay(o),o=rs;ls;o=ls);return o;}
```

3.9 主席树

```
//询问区间第k小
int bt(int g,int l,int r,int x){
    int mid=(l+r)>>1,now=++t0;t[now]=t[g],t[now].x++;
    if(l==r)return now;
    if(x<=mid)t[now].l=bt(t[g].l,l,mid,x);
    else t[now].r=bt(t[g].r,mid+1,r,x);
    return now;
}
int query(int rg,int lg,int x,int l,int r){
    if(l==r)return l;
    int mid=(l+r)>>1,v=t[t[rg].l].x-t[t[lg].l].x;
    if(v>=x)return query(t[rg].l,t[lg].l,x,l,mid);
    else return query(t[rg].r,t[lg].r,x-v,mid+1,r);
}
```

3.10 树链剖分

```
int et,la[N],n,q,id[N],fa[N],son[N],siz[N],dep[N],top[N],x,y,i,dfn;
#define swap(x,y) (st=x,x=y,y=st)
struct E{int to,nxt;}e[N<<1];
#define add(x,y) (e[++et]=(E){y,la[x]},la[x]=et)
#define max(a,b) (a>b?a:b)
void dfs(int x,int f){
    dep[x]=dep[fa[x]=f]+1,siz[x]=1;
    for(int i=la[x];i;i=e[i].nxt)if(e[i].to!=f){
        dfs(e[i].to,x),siz[x]+=siz[e[i].to];
        if(siz[e[i].to]>siz[son[x]])son[x]=e[i].to;
    }
}
void dfs2(int x,int t){
    if(id[x]=++dfn,top[x]=t,son[x])dfs2(son[x],t);
    for(int i=la[x];i;i=e[i].nxt)
    if(e[i].to!=fa[x]&&e[i].to!=son[x])dfs2(e[i].to,e[i].to);
}
void query(int l,int r){
    //询问线段树中区间[l,r]的答案
```

```
}
void getlca(int x,int y){
    int fx=top[x],fy=top[y];
    while(fx!=fy){
        if(dep[fx]<dep[fy])swap(fx,fy),swap(x,y);
        query(id[fx],id[x]),x=fa[fx],fx=top[x];
    }
    if(dep[x]>dep[y])swap(x,y);
    query(id[x],id[y]);
}
```

3.11 LCT

```
#define cmax(a,b) (a<b?a=b:1)
const int N=30010;
int n,q,u,v,fa[N],son[N][2],val[N],sum[N],max[N],swp,rev[N];
#define swap(a,b) (swp=a,a=b,b=swp)
void pu(int t){
    max[t]=val[t],cmax(max[t],max[son[t][0]]),cmax(max[t],max[son[t][1]]);
    sum[t]=sum[son[t][0]]+sum[son[t][1]]+val[t];
}
void pd(int t){
    if(rev[t])rev[t]=0,rev[son[t][0]]^=1,rev[son[t][1]]^=1,swap(son[t][0],
        son[t][1]);
}
bool nr(int t){return son[fa[t]][0]==t||son[fa[t]][1]==t;}
void rtt(int t,int f=0,bool p=0){
    p=son[f=fa[t]][1]==t,
    fa[t]=fa[f],nr(f)?son[fa[f]][son[fa[f]][1]==f]=t:1,
    (son[f][p]=son[t][!p])?fa[son[f][p]]=f:1,
    pu(son[fa[f]=t][!p]=f);
}
void pv(int t){
    if(nr(t))pv(fa[t]);pd(t);
}
void splay(int t,int f=0){
    for(pv(t);nr(t);rtt(t))
    if(nr(f=fa[t]))rtt(son[f][1]==t^son[fa[f]][1]==f?t:f);
    pu(t);
}
void access(int t,int las=0){
    for(;t;splay(t),son[t][1]=las,las=t,t=fa[t]);
}
void makeroot(int t){
    access(t),splay(t),rev[t]^=1;
}
void link(int u,int v){
    makeroot(u),fa[u]=v;
}
```

```
void cut(int u,int v){
    makeroot(u),access(v),splay(v),son[v][0]=fa[u]=0;
}
```

3.12 KD-Tree

```
int n,m,op,x,y,i,dd,rt,ans;
struct T{int d[2],s[2],x[2],y[2];}t[1<<20];
struct P{int d[2];}a[1<<19];
bool operator<(const P&a,const P&b){return a.d[dd]<b.d[dd]||a.d[dd]==b.d[dd
    ]&&a.d[dd^1]<b.d[dd^1];}
#define abs(x) (x>0?x:-(x))
#define max(a,b) (a>b?a:b)
#define cmax(a,b) (a<b?a=b:a)
#define cmin(a,b) (a>b?a=b:a)
#define ls t[now].s[0]
#define rs t[now].s[1]
void mt(int f,int x){
    cmin(t[f].x[0],t[x].x[0]),cmax(t[f].x[1],t[x].x[1]);
    cmin(t[f].y[0],t[x].y[0]),cmax(t[f].y[1],t[x].y[1]);
}
int bt(int l,int r,int d){
    dd=d;int now=l+r>>1;
    std::nth_element(a+l,a+now,a+r+1);
    t[now].d[0]=t[now].x[0]=t[now].x[1]=a[now].d[0];
    t[now].d[1]=t[now].y[0]=t[now].y[1]=a[now].d[1];
    if(l<now)ls=bt(l,now-1,d^1),mt(now,ls);
    if(now<r)rs=bt(now+1,r,d^1),mt(now,rs);
    return now;
}
int getdis(int p){
    return max(t[p].x[0]-x,0)+max(x-t[p].x[1],0)+max(t[p].y[0]-y,0)+max(y-t[
        p].y[1],0);
}
void ins(int n){
    for(int p=rt,dd=0;p;dd^=1){
        mt(p,n);int&nx=t[p].s[t[n].d[dd]>=t[p].d[dd]];
        if(nx==0){nx=n;return;}else p=nx;
    }
}
void query(int now){
    int d[2]={1<<30,1<<30},d0=abs(t[now].d[0]-x)+abs(t[now].d[1]-y),p;
    if(ls)d[0]=getdis(ls);if(rs)d[1]=getdis(rs);p=d[0]>=d[1];cmin(ans,d0);
    if(d[p]<ans)query(t[now].s[p]);
    if(d[p^1]<ans)query(t[now].s[p^1]);
}
```

4 DP

4.1 斜率优化

```
//q表示单调队列，当f[i]=max()时维护开口向下的凸壳，斜率递减/f[i]=min()时维护开口向上的凸壳，斜率递增
//大不了暴力枚举不等号方向跟暴力拍一下，反正就四种情况
for(i=0;i<=n;q[++r]=i++){
        while(l<r&&y[q[l+1]]-y[q[l]]<=(cnt[q[l]+1]-cnt[q[l+1]+1])*d[i])l++;//未移项的不等式
        j=q[l],f[i]=f[j]+s[i]+sum[j+1]-sum[i]-(dis-d[i])*(cnt[j+1]-cnt[i]),y[i]=f[i]+sum[i+1]-dis*cnt[i+1];//表达式
        while(l<r&&(y[q[r]]-y[q[r-1]])*(cnt[i+1]-cnt[q[r]+1])<=(y[i]-y[q[r]])*(cnt[q[r]+1]-cnt[q[r-1]+1]))r--;//比较斜率
}
```

5 图论

5.1 边表

```
int et=1,la[N];
struct E{int to,v,nxt;}e[N<<1];
void add(int x,int y,int v){
    e[++et]=(E){y,v,la[x]},la[x]=et;
}
```

5.2 spfa

```
int q[N],l,r,d[N],in[N];
void spfa(int s){
    memset(d,63,sizeof(d));
    for(q[l=r=0]=s,d[s]=0,in[s]=1;l<=r;in[q[l++]]=0)
    for(int i=la[q[l]];i;i=e[i].nxt)
    if(d[e[i].to]>d[q[l]]+e[i].v){
        d[e[i].to]=d[q[l]]+e[i].v;
        if(!in[e[i].to])in[q[++r]=e[i].to]=1;
    }
}
```

5.3 Dijkstra

```
int l,r,in[M];ld d[M];
struct Q{
  int key;ld value;
  bool operator<(const Q&a)const{return value>a.value;}
};
```

```
std::priority_queue<Q>q;
ld dijkstra(){
  for(int i=2;i<=idt;i++)d[i]=1e10;
  for(q.push((Q){1,d[1]=0});!q.empty();){
    Q tmp=q.top();q.pop();
    if(in[tmp.key])continue;in[tmp.key]=1;
    for(int i=la[tmp.key];i;i=e[i].nxt)
    if(d[e[i].to]>d[tmp.key]+e[i].v)
      q.push((Q){e[i].to,d[e[i].to]=d[tmp.key]+e[i].v});
  }
  return d[2];
}
```

5.4 二分图匹配-匈牙利

```
int dfs(int x){
    for(int i=la[x];i;i=e[i].nxt)
    if(vis[e[i].to]!=tim){
        vis[e[i].to]=tim;
        if(!fa[e[i].to]||dfs(fa[e[i].to]))
        return fa[e[i].to]=x,1;
    }
    return 0;
}
main(){...;for(int i=1;i<=n;i++)tim++,ans+=dfs(i);...;}
```

5.5 Dinic

```
#include<cstdio>
#include<cstring>
#define N 5010
int n,m,s,t,la[N],et=1,d[N],cur[N],q[N],l,r;
struct E{int to,flow,nxt;}e[100000];
void add(int x,int y,int v){
    e[++et]=(E){y,v,la[x]},la[x]=et;
    e[++et]=(E){x,0,la[y]},la[y]=et;
}
int bfs(){
    memset(d,0,sizeof(d));
    for(q[l=r=0]=t,d[t]=1;l<=r;l++)
    for(int i=la[q[l]];i;i=e[i].nxt)
    if(e[i^1].flow&&!d[e[i].to])
    d[q[++r]=e[i].to]=d[q[l]]+1;
    return d[s];
}
int dfs(int x,int ret){
    if(x==t||ret==0)return ret;
    int tmp,flow=0;
```

```
    for(int&i=cur[x];i;i=e[i].nxt)
    if(d[x]==d[e[i].to]+1){
        tmp=dfs(e[i].to,e[i].flow<ret-flow?e[i].flow:ret-flow);
        e[i].flow-=tmp,e[i^1].flow+=tmp,flow+=tmp;
        if(ret==flow)return flow;
    }
    return flow;
}
int maxflow(){
    int flow=0;
    while(bfs())memcpy(cur,la,sizeof(la)),flow+=dfs(s,1<<30);
    return flow;
}
```

5.6 MCMF

```
#include<cstdio>
#include<cstring>
const int N=2000,M=40000,oo=1<<28;
int p[N],d[N],q[1<<20],la[N],s,t,et=1,in[N];
struct E{int to,flow,cost,nxt;}e[M<<1];
#define cmin(a,b) (a>b?a=b:1)
void add(int from,int to,int flow,int cost){
    e[++et]=(E){to,flow,cost,la[from]},la[from]=et;
    e[++et]=(E){from,0,-cost,la[to]},la[to]=et;
}
int spfa(){
    memset(d,63,sizeof(d));int l,r,i;
    for(q[l=r=1]=s,in[s]=1,d[s]=0;l<=r;in[q[l++]]=0)
    for(i=la[q[l]];i;i=e[i].nxt)
    if(e[i].flow&&d[e[i].to]>d[q[l]]+e[i].cost){
        d[e[i].to]=d[q[l]]+e[i].cost;p[e[i].to]=i;
        if(!in[e[i].to])in[q[++r]=e[i].to]=1;
    }
    return d[t]<d[0];
}
int mincost(){
    int flow=0,cost=1<<20,u,tmp;
    while(spfa()){tmp=oo;
        for(u=t;u!=s;u=e[p[u]^1].to)cmin(tmp,e[p[u]].flow);
        for(u=t;u!=s;u=e[p[u]^1].to)e[p[u]].flow-=tmp,e[p[u]^1].flow+=tmp;
        flow+=tmp,cost+=d[t]*tmp;
    }
    return cost;
}
```

5.7 LCA-倍增

```
int lca(int x,int y){
    int k,i;
    if(dep[x]<dep[y])k=x,x=y,y=k;
    for(i=0,k=dep[x]-dep[y];k;k>>=1,i++)
    if(k&1)x=fa[x][i];
    if(x==y)return x;
    for(i=0;fa[x][i]!=fa[y][i];i++);
    for(i--;~i;i--)if(fa[x][i]!=fa[y][i])x=fa[x][i],y=fa[y][i];
    return fa[x][0];
}
```

5.8 LCA-树链剖分版

```
//前面两个dfs跟树剖一样
int lca(int x,int y){
    while(top[x]!=top[y])
    if(dep[top[x]]>dep[top[y]])x=fa[top[x]];else y=fa[top[y]];
    return dep[x]<dep[y]?x:y;
}
```

5.9 带花树

```
#define N 1010
int n,m,x,y,v,mate[N],fa[N],pre[N],la[N],et,ans,q[N],l,r,sta[N],vis[N],tim;
struct E{int to,nxt;}e[6010];
#define add(x,y) (e[++et]=(E){y,la[x]},la[x]=et)
int gf(int x){return fa[x]==x?x:fa[x]=gf(fa[x]);}
int lca(int x,int y){
    for(++tim,x=gf(x),y=gf(y);;v=x,x=y,y=v)if(x){
        if(vis[x]==tim)return x;
        vis[x]=tim,x=gf(pre[mate[x]]);
    }
}
int blossom(int x,int y,int g){
    while(gf(x)!=g){
        if(pre[x]=y,sta[mate[x]]==1)sta[q[++r]=mate[x]]=0;
        if(gf(x)==x)fa[x]=g;if(gf(mate[x])==mate[x])fa[mate[x]]=g;
        y=mate[x],x=pre[y];
    }
}
int match(int s){int i,j,las;
    memset(pre,0,sizeof(pre));
    memset(sta,-1,sizeof(sta));
    for(i=1;i<=n;i++)fa[i]=i;
    for(q[l=r=0]=s,sta[s]=0;l<=r;l++)
    for(i=la[q[l]];i;i=e[i].nxt)
    if(sta[e[i].to]==-1){
```

```
        pre[e[i].to]=q[l],sta[e[i].to]=1;
        if(!mate[e[i].to]){
            for(j=q[l],i=e[i].to;j;j=pre[i=las])las=mate[j],mate[j]=i,mate[i
                ]=j;
            return 1;
        }
        sta[q[++r]=mate[e[i].to]]=0;
    }
    else if(gf(e[i].to)!=gf(q[l])&&sta[e[i].to]==0)
        j=lca(e[i].to,q[l]),blossom(e[i].to,q[l],j),blossom(q[l],e[i].to,j);
    return 0;
}
```

6 字符串

6.1 Trie

```
void add(char c){
    if(!ch[las][c-='a'])ch[las][c]=++tot;
    las=ch[las][c];
}
```

6.2 KMP

```
for(i=1;s[i];f[i+1]=s[i]==s[j]?j+1:0,i++)
for(j=f[i];j&&s[i]!=s[j];j=f[j]);
```

6.3 AC 自动机

```
//ch是建好的Trie
void build_AC(){
    for(q[l=r=0]=0;l<=r;l++)
    for(int i=0,j;i<26;i++)
    if(j=ch[q[l]][i]){
        if(q[++r]=j,q[l])fail[j]=ch[fail[q[l]]][i];
    }
    else ch[q[l]][i]=ch[fail[q[l]]][i];
}
```

6.4 回文自动机

```
int nn(int l){return len[tot]=l,tot++;}//new_node
int gf(int x){//get_fail
    while(s[n-1-len[x]]!=s[n])x=fail[x];
    return x;
```

```
}
void add(char c){
    if(las=gf(las),!ch[las][c-='a']){
        now=nn(len[las]+2);
        fail[now]=ch[gf(fail[las])][c];
        ch[las][c]=now;
    }
    las=ch[las][c],f[las]++;
}
```

6.5 SA 带 height

```
#define cmp(u,v) (x[u]!=x[v]||x[u+k]!=x[v+k])
for(i=1;i<=n;i++)c[x[i]=s[i]-'a'+1]++;
for(i=1;i<=m;i++)c[i]+=c[i-1];
for(i=n;i;i--)sa[c[x[i]]--]=i;
for(k=1;k<n&&(k==1||m<n);k<<=1,T=x,x=y,y=T){
    for(yt=0,i=n-k+1;i<=n;i++)y[++yt]=i;
    for(i=1;i<=n;i++)if(sa[i]>k)y[++yt]=sa[i]-k;
    for(i=1;i<=m;i++)c[i]=0;
    for(i=1;i<=n;i++)c[x[i]]++;
    for(i=1;i<=m;i++)c[i]+=c[i-1];
    for(i=n;i;i--)sa[c[x[y[i]]]--]=y[i];
    for(m=0,i=1;i<=n;i++)y[sa[i]]=i==1||cmp(sa[i],sa[i-1])?++m:m;
}
for(i=1;i<=n;i++)rk[sa[i]]=i;
for(i=1,k=0;i<=n;hei[rk[i++]]=k)
for(k?k--:0,j=sa[rk[i]-1];s[i+k]==s[j+k];k++);
```

6.6 SAM

```
class SAM{public:
    int S,tot,las,fail[N],ch[N][26],cnt,len[N];
    SAM(){S=tot=las=1;}
    void add(char c){
        int p=las,np=++tot;
        for(len[np]=len[p]+1;p&&!ch[p][c];p=fail[p])ch[p][c]=np;
        if(las=np,!p)fail[np]=S;
        else if(len[p]+1==len[ch[p][c]])fail[np]=ch[p][c];
        else{
            int q=ch[p][c],r=++tot;
            len[r]=len[p]+1,fail[r]=fail[q],fail[q]=fail[np]=r;
            memcpy(ch[r],ch[q],sizeof(ch[q]));
            for(;p&&ch[p][c]==q;p=fail[p])ch[p][c]=r;
        }
    }
}sam;
```

7 数学

7.1 快速幂

```
int power(int t,int k,int p){
    int f=1;
    for(;k;k>>=1,t=1LL*t*t%p)
    if(k&1)f=1LL*f*t%p;
    return f;
}
```

7.2 exgcd

```
void exgcd(int a,int b,int&x,int&y){!b?x=1,y=0:(exgcd(b,a%b,y,x),y-=a/b*x);}
```

7.3 线性求逆元

```
inv[1]=1;for(int i=2;i<=n;i++)inv[i]=1LL*(p-p/i)*inv[p%i]%p;
```

7.4 线性筛求 φ

```
for(i=2;i<=n;i++)
for(vis[i]==0?p[++t]=i,phi[i]=i-1:1,j=1;j<=t&&i*p[j]<=n;j++){
    vis[i*p[j]]=1;
    if(i%p[j]==0){
        phi[i*p[j]]=phi[i]*p[j];
        break;
    }
    phi[i*p[j]]=phi[i]*(p[j]-1);
}
```

7.5 Miller-Rabin + Rho

```
#include<cstdio>
#include<cstdlib>
typedef long long ll;
ll _,n,x,ans,st;
ll gcd(ll x,ll y){return y==0?x:gcd(y,x%y);}
#define abs(x) (x>0?x:-(x))
#define cmax(a,b) (a<b?a=b:1)
ll mul(ll a,ll b,ll p){
    ll tmp=(a*b-(ll)((long double)a/p*b+1e-7)*p);
    return tmp<0?tmp+p:tmp;
}
ll power(ll t,ll k,ll p){
    ll f=1;
```

```
14      for(;k;k>>=1,t=mul(t,t,p))if(k&1)f=mul(f,t,p);
15      return f;
16  }
17  bool check(ll a,int k,ll p,ll q){
18      ll t=power(a,q,p);
19      if(t==1||t==p-1)return 1;
20      for(;k--;){
21          t=mul(t,t,p);
22          if(t==p-1)return 1;
23      }
24      return 0;
25  }
26  bool mr(ll p){
27      if(p<=1)return 0;
28      if(p==2)return 1;
29      if(~p&1)return 0;
30      ll q=p-1;int i,k=0;
31      while(~q&1)q>>=1,k++;
32      for(i=0;i<5;i++)
33      if(!check(rand()%(p-1)+1,k,p,q))return 0;
34      return 1;
35  }
36  ll rho(ll n,ll c){
37      ll x=rand()%n,y=x,p=1;
38      while(p==1)
39          x=(mul(x,x,n)+c)%n,
40          y=(mul(y,y,n)+c)%n,
41          y=(mul(y,y,n)+c)%n,
42          p=gcd(n,abs(x-y));
43      return p;
44  }
45  void solve(ll n){
46      if(n==1)return;
47      if(mr(n)){cmax(ans,n);return;}
48      if(~n&1)cmax(ans,2),solve(n>>1);
49      else{
50          ll t=n;
51          while(t==n)t=rho(n,rand()%(n-1)+1);
52          solve(t),solve(n/t);
53      }
54  }
55  int main(){
56      for(srand(1626),scanf("%lld",&_);_--;){
57          scanf("%lld",&x),ans=0;solve(x);
58          if(ans==x)puts("Prime");
59          else printf("%lld\n",ans);
60      }
61  }
```

7.6 高斯消元

```
void solve(int n){
    int i,j,k,las;double t;
    for(i=1;i<=n;i++){
        for(t=0,j=i;j<=n;j++)
        if(abs(a[j][i])>t)t=abs(a[j][i]),las=j;
        if(j=las,j!=i)for(k=1;k<=n+1;k++)std::swap(a[i][k],a[j][k]);
        for(j=i+1;j<=n;j++)
        for(t=a[j][i]/a[i][i],k=i;k<=n+1;k++)a[j][k]-=a[i][k]*t;
    }
    for(i=n;i>=1;i--)
    for(a[i][n+1]/=a[i][i],j=i-1;j;j--)a[j][n+1]-=a[j][i]*a[i][n+1];
}
```

7.7 矩阵快速幂

```
struct M{int m[110][110];}c,t,f;
M operator*(const M&a,const M&b){
    static long long tp[110][110];
    memset(tp,0,sizeof(tp));
    for(int i=0;i<k;i++)
    for(int kk=0;kk<k;kk++)
    for(int j=0;j<k;j++)
    tp[i][j]+=1LL*a.m[i][kk]*b.m[kk][j];
    for(int i=0;i<k;i++)
    for(int j=0;j<k;j++)c.m[i][j]=tp[i][j]%p;
    return c;
}
M power(M t,int k){
    for(f=t,k--;k;k>>=1,t=t*t)if(k&1)f=f*t;
    return f;
}
```

7.8 k 阶线性递推-特征多项式

```
#include<cstdio>
#define p 1000000007
long long c[4010];
int n,k,u[2010],ans,x;int cnt;
struct P{int s[2010];}f,t;
void mult(P&a,const P&b){
    for(int i=0;i<k+k-1;i++)c[i]=0;
    for(int i=0;i<k;i++)
    for(int j=0;j<k;j++){
        c[i+j]+=1LL*a.s[i]*b.s[j];
        if(c[i+j]>=1LL<<62)c[i+j]%=p;
    }
```

```
    for(int i=k+k-2;~i;i--)
    if(c[i]%=p,i>=k){
        for(int j=0;j<k;j++){
            c[i-1-j]+=c[i]*u[j];
            if(c[i-1-j]>=1LL<<62)c[i-1-j]%=p;
        }
        c[i]=0;
    }
    for(int i=0;i<k;i++)a.s[i]=c[i];
}
int main(){
//hn = a1*h(n-1) + a2*h(n-2) + ... + ak*h(n-k)
    scanf("%d%d",&n,&k);
    for(int i=0;i<k;i++)scanf("%d",u+i),u[i]%=p,u[i]<0?u[i]+=p:1;//a
    for(t.s[1]=f.s[0]=1;n;n>>=1,mult(t,t))if(n&1)mult(f,t);
    for(int i=0;i<k;i++)scanf("%d",&x),x%=p,x<0?x+=p:1,ans=(ans+1LL*x*f.s[i
        ])%p;
    printf("%d\n",ans);
}
```

7.9 拉格朗日插值

```
#include<cstdio>
#define p 1004535809
#define N 3010
int n,x[N],y[N],q,l,r,x0,sum[N][N],isum[N][N],inv[250010],L[N],R[N],ans;
#define getinv(x) (x>=0?inv[x]:p-inv[-(x)])
int main(){
    scanf("%d",&n);inv[1]=1;
    for(int i=1;i<=n;i++)scanf("%d%d",x+i,y+i);
    for(int i=2;i<=250000;i++)inv[i]=1LL*(p-p/i)*inv[p%i]%p;
    for(int i=1;i<=n;i++){
        sum[i][0]=isum[i][0]=1;
        for(int j=1;j<i;j++)sum[i][j]=1LL*sum[i][j-1]*(p+x[i]-x[j])%p,isum[i
            ][j]=1LL*isum[i][j-1]*getinv(x[i]-x[j])%p;
        sum[i][i]=sum[i][i-1],isum[i][i]=isum[i][i-1];
        for(int j=i+1;j<=n;j++)sum[i][j]=1LL*sum[i][j-1]*(p+x[i]-x[j])%p,
            isum[i][j]=1LL*isum[i][j-1]*getinv(x[i]-x[j])%p;
    }
    for(scanf("%d",&q);q--;printf("%d\n",ans)){
        scanf("%d%d%d",&l,&r,&x0);L[l-1]=R[r+1]=1;ans=0;
        for(int i=l;i<=r;i++)L[i]=1LL*L[i-1]*(p+x0-x[i])%p;
        for(int i=r;i>=l;i--)R[i]=1LL*R[i+1]*(p+x0-x[i])%p;
        for(int i=l;i<=r;i++)ans=(ans+1LL*sum[i][l-1]*isum[i][r]%p*L[i-1]%p*
            R[i+1]%p*y[i])%p;
    }
}
```

7.10 牛顿插值

```
#include<cstdio>
#define N 3010
#define p 1004535809
int n,q,x[N],f[N][N],inv[250010];
#define getinv(x) (x>=0?inv[x]:p-inv[-(x)])
int main(){
    scanf("%d",&n);inv[1]=1;
    for(int i=2;i<=250000;i++)inv[i]=1LL*(p-p/i)*inv[p%i]%p;
    for(int i=1;i<=n;i++)scanf("%d%d",x+i,f[i]+1);
    for(int i=2;i<=n;i++)
    for(int j=2,k=i-1;j<=i;j++,k--)f[i][j]=1LL*(p+f[i-1][j-1]-f[i][j-1])*
        getinv(x[k]-x[i])%p;
    scanf("%d",&q);
    for(int l,r,x0,ans;q--;printf("%d\n",ans)){
        scanf("%d%d%d",&l,&r,&x0);ans=0;
        for(int i=l,j=1,sum=1;i<=r;i++,j++)ans=(ans+1LL*f[i][j]*sum)%p,sum=1
            LL*sum*(p+x0-x[i])%p;
    }
}
```

7.11 BSGS

```
//y^x==z (mod p) ->x=?
scanf("%d%d%d",&y,&z,&p),y%=p,z%=p;j=z;
if(y==0){puts("Cannot␣find␣x");continue;}
for(k=s=1;k*k<=p;k++);
std::map<int,int>hash;flag=0;
for(int i=0;i<k;i++,s=1LL*s*y%p,j=1LL*j*y%p)hash[j]=i;
for(int i=1,j=s;i<=k&&!flag;i++,j=1LL*j*s%p)
if(hash.count(j))ans=i*k-hash[j],flag=1;
if(flag==0)puts("Cannot␣find␣x");
else printf("%d\n",ans);
```

7.12 阶、原根

```
bool check(){
    for(i=2;i*i<=p;i++)
    if((p-1)%i==0&&power(g,(p-1)/i,p)==1)return 0;
    return 1;
}
void getroot(){
    if(p==2)g=1;else for(g=2;!check();g++);
    for(ind[1]=0,pw[0]=i=1;i<p-1;i++)pw[i]=pw[i-1]*g%p,ind[pw[i]]=i;
}
```

7.13 exBSGS

```
int bsgs(int a,ll b,int p){
    if(a%=p,b%=p,b==1)return 0;
    ll t=1;int f,g,delta=0,m=sqrt(p)+1,i;
    for(g=gcd(a,p);g!=1;g=gcd(a,p)){
        if(b%g)return -1;
        b/=g,p/=g,t=t*(a/g)%p,delta++;
        if(b==t)return delta;
    }
    std::map<int,int>hash;
    for(i=0;i<m;i++,b=b*a%p)hash[b]=i;
    for(i=1,f=power(a,m);i<=m+1;i++)
    if(t=t*f%p,hash.count(t))return i*m-hash[t]+delta;
    return -1;
}
```

7.14 $O(n^{2/3})$ 求 φ/μ 的前缀和

```
#include<cstdio>
#include<cmath>
typedef long long ll;
typedef unsigned U;
const ll oo=1LL<<60;
#define N 1<<22
const int M=(1<<18)-1;
int n,k,p[N],t,vis[N],T,a[20];ll phi[N],miu[N];
class map{public:
    int et,la[M+1];
    struct E{int nxt;U x;ll ans;}e[1<<18];
    inline ll find(U x){
        for(int i=la[x&M];i;i=e[i].nxt)
        if(e[i].x==x)return e[i].ans;
        return -oo;
    }
    inline void ins(U x,ll ans){
        e[++et]=(E){la[x&M],x,ans},la[x&M]=et;
    }
}_phi,_miu;
ll getphi(U n){
    if(n<=k)return phi[n];
    ll ans=_phi.find(n);
    if(ans!=-oo)return ans;ans=n*(n+1LL)/2;
    for(U l=2,r;l<=n;l=r+1)r=n/(n/l),ans-=(r-l+1)*getphi(n/l);
    return _phi.ins(n,ans),ans;
}
ll getmiu(U n){
    if(n<=k)return miu[n];
    ll ans=_miu.find(n);
```

```
    if(ans!=-oo)return ans;ans=1;
    for(U l=2,r;l<=n;l=r+1)r=n/(n/l),ans-=(r-l+1)*getmiu(n/l);
    return _miu.ins(n,ans),ans;
}
int main(){
    scanf("%d",&T);
    for(int i=1;i<=T;i++)scanf("%d",a+i),n<a[i]?n=a[i]:1;
    k=2.5*pow(n,2.0/3)+1;phi[1]=miu[1]=1;
    for(int i=2;i<=k;i++){
        if(!vis[i])p[++t]=i,phi[i]=i-1,miu[i]=-1;
        for(int j=1;j<=t&&i*p[j]<=k;j++){
            vis[i*p[j]]=1;
            if(i%p[j]==0){
                phi[i*p[j]]=phi[i]*p[j];
                break;
            }
            phi[i*p[j]]=phi[i]*phi[p[j]],miu[i*p[j]]=-miu[i];
        }
    }
    for(int i=2;i<=k;i++)phi[i]+=phi[i-1],miu[i]+=miu[i-1];
    for(int i=1;i<=T;i++)printf("%lld␣%lld\n",getphi(a[i]),getmiu(a[i]));
}
```

7.15 单纯形

```
struct LP{
//Ax<=b, max(cx), x>=0
    int n,m;ld a[10010][1010],b[10010],c[1010],v;
    void setup(int _n,int _m){
        n=_n,m=_m,v=0;
    }
    void pivot(int l,int e){
        int i,j;
        for(j=1;j<=n;j++)if(j!=e)a[l][j]/=a[l][e];
        b[l]/=a[l][e],a[l][e]=1/a[l][e];
        for(i=1;i<=m;i++)
        if(i!=l&&std::fabs(a[i][e])>eps){
            for(j=1;j<=n;j++)if(j!=e)a[i][j]-=a[i][e]*a[l][j];
            b[i]-=a[i][e]*b[l],a[i][e]*=-a[l][e];
        }
        for(j=1;j<=n;j++)if(j!=e)c[j]-=c[e]*a[l][j];
        v+=c[e]*b[l],c[e]*=-a[l][e];
    }
    ld simplex(){
        int i,l,e;ld tmp;
        while(1){tmp=eps;e=n+1;
            for(i=1;i<=n;i++)if(c[i]>tmp)tmp=c[i],e=i;
            if(e==n+1)return v;
            tmp=oo;
```

```
            for(i=1;i<=m;i++)
            if(a[i][e]>eps&&b[i]/a[i][e]<tmp)tmp=b[i]/a[i][e],l=i;
            if(tmp==oo)return tmp;
            pivot(l,e);
        }
    }
}dcx;
```

7.16 FFT

```
for(k=1;k<n<<1;k<<=1,L++);L--;
for(i=1;i<k;i++)rev[i]=(rev[i>>1]>>1)|((i&1)<<L);
for(i=0;i<=k;i++)w[1][k-i]=w[0][i]=(P){cos(PI*2*i/k),sin(PI*2*i/k)};
void FFT(P*a,int n,P*w){
    int i,j,k;
    for(i=1;i<n;i++)if(i>rev[i])std::swap(a[i],a[rev[i]]);
    for(i=2;i<=n;i<<=1)
    for(j=0;j<n;j+=i)
    for(k=0;k<i>>1;k++)
    tmp=a[j+k+i/2]*w[n/i*k],a[j+k+i/2]=a[j+k]-tmp,a[j+k]=a[j+k]+tmp;
}
```

7.17 NTT

```
#define ck(x) (x>=p?x-=p:1)
for(n=1;n<m;n<<=1,l++);n<<=1;g=power(3,(P-1)/n);
for(w[0][0]=w[1][0]=i=1;i<n;i++){
    w[1][n-i]=w[0][i]=g*w[0][i-1]%P;
    rev[i]=(rev[i>>1]>>1)|((i&1)<<l);
}

void FFT(int*a,int*w){
    int i,j,k,tmp;
    for(i=0;i<n;i++)if(i>rev[i])tmp=a[i],a[i]=a[rev[i]],a[rev[i]]=tmp;
    for(i=2;i<=n;i<<=1)
    for(j=0;j<n;j+=i)
    for(k=0;k<i>>1;k++)
    tmp=1LL*a[j+k+i/2]*w[n/i*k]%p,a[j+k+i/2]=a[j+k]-tmp+p,a[j+k]+=tmp,ck(a[j
        +k+i/2]),ck(a[j+k]);
}
```

7.18 多项式求逆

```
#include<cstdio>
#include<algorithm>
#define N 1<<18|10
#define p 1004535809
```

```
int n,l,k,w[2][N],rev[N],a[N],b[N],c[N],g,tot=1<<18;
int power(int t,int k){
    int f=1;
    for(;k;k>>=1,t=1LL*t*t%p)if(k&1)f=1LL*f*t%p;
    return f;
}
#define ck(x) (x>=p?x-=p:1)
void FFT(int*a,int n,int*w){
    int i,j,k,l,tmp;
    for(i=1;i<n;i++)if(rev[i]<i)tmp=a[rev[i]],a[rev[i]]=a[i],a[i]=tmp;
    for(i=2;i<=n;i<<=1)
    for(l=i>>1,j=0;j<n;j+=i)
    for(k=0;k<l;k++)
    tmp=1LL*a[j+k+l]*w[tot/i*k]%p,a[j+k+l]=a[j+k]-tmp+p,a[j+k]+=tmp,ck(a[j+k
        +l]),ck(a[j+k]);
}
void getinv(int deg,int*a,int*b){
    if(deg==1){
        b[0]=power(a[0],p-2);
        return;
    }
    getinv(deg+1>>1,a,b);
    for(l=0,k=1;k<deg;k<<=1,l++);k<<=1;
    for(int i=0;i<k;i++)rev[i]=(rev[i>>1]>>1)|((i&1)<<l);
    std::copy(a,a+deg,c);
    std::fill(c+deg,c+k,0);
    FFT(b,k,w[0]),FFT(c,k,w[0]);
    for(int i=0;i<k;i++)b[i]=(2-1LL*c[i]*b[i]%p)*b[i]%p+p,ck(b[i]);
    FFT(b,k,w[1]);g=power(k,p-2);
    for(int i=0;i<k;i++)b[i]=1LL*b[i]*g%p;
    std::fill(b+deg,b+k,0);
}
int main(){
    w[0][0]=w[1][0]=1;g=power(3,(p-1)/tot);
    for(k=1;k<=tot;k++)w[0][k]=w[1][tot-k]=1LL*g*w[0][k-1]%p;
    scanf("%d",&n);
    for(int i=0;i<n;i++)scanf("%d",a+i);
    getinv(n,a,b);
    for(int i=0;i<n;i++)printf("%d␣",b[i]);puts("");
}
```

7.19 找规律

```
#include<cstdio>
typedef long long ll;
const int MAXS=1000;
double a[MAXS][MAXS+1],temp;
void swap(double&i,double&j){temp=i;i=j;j=temp;}
double abs(double a){return a>0?a:-a;}
```

```
void solve(ll n){
    ll i,j,k,lasi;double t,maxi;
    for(i=1;i<=n;i++){
        maxi=0;
        for(j=i;j<=n;j++)
        if(abs(a[j][i])>maxi){
            maxi=abs(a[j][i]);lasi=j;
        }j=lasi;
        if(j!=i)for(k=1;k<=n+1;k++)swap(a[i][k],a[j][k]);
        for(j=i+1;j<=n;j++){
            t=a[j][i]/a[i][i];
            for(k=i;k<=n+1;k++)a[j][k]-=a[i][k]*t;
        }
    }
    for(i=n;i>=1;i--){
        a[i][n+1]/=a[i][i];
        for(j=i-1;j>=1;j--)
            a[j][n+1]-=a[j][i]*a[i][n+1];
    }
}
ll f[1000],n,k,ans,F[1000]={3,9,13,25,81,225,477,1089,2785,6889,
                16237,38809,94641,229441,551613,1329409,3215041};
int main(){
    int i,j,k;double ans;n=15;
    for(int l=2;l<=n;l++){
        for(i=1;i<=l;i++)
        for(j=i;j<=i+l;j++)
        a[i][j-i+1]=F[j];
//  for(i=1;i<=l;i++){for(j=1;j<=l+1;j++)printf("%.0lf ",a[i][j]);puts("");}
        solve(l);
        for(i=1;i<=l;i++)printf("%.5lf␣",a[i][l+1]);printf("\n");
        getchar();
    }
}
```

8 计算几何

8.1 Point 的定义（在复数域上）

```
typedef double ld;
#define PP const P&
struct P{
    ld x,y;
    bool operator<(PP a)const{return x<a.x||x==a.x&&y<a.y;}
    P operator+(PP a)const{return(P){x+a.x,y+a.y};}
    P operator-(PP a)const{return(P){x-a.x,y-a.y};}
    P operator*(ld a)const{return(P){x*a,y*a};}
    P operator/(ld a)const{return(P){x/a,y/a};}
```

```
    P operator*(PP a)const{return(P){x*a.x-y*a.y,x*a.y+y*a.x};}
    ld operator|(PP a)const{return x*a.x+y*a.y;}
    ld operator&(PP a)const{return x*a.y-y*a.x;}
    P operator/(PP a)const{ld g=a.x*a.x+a.y*a.y;return (P){(x*a.x+y*a.y)/g,(y*a.x-x*a.y)/g};}
    void _sqrt(){ld t=atan2(y,x)*.5,len=sqrt(sqrt(x*x+y*y));x=len*cos(t),y=len*sin(t);}
};
ld len2(PP a){return a.x*a.x+a.y*a.y;}
#define check(a,b,c) ((b-a)&(c-a))
```

8.2 二维凸包

```
std::sort(a+1,a+1+n);
for(i=1;i<=n;i++)a[n+n-i+1]=a[i];
for(r=0,i=1;i<=n+n;q[++r]=a[i++])
while(r>1&&check(q[r-1],q[r],a[i])<=0)r--;
```

8.3 三维凸包

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<cmath>
typedef double ld;
char ch,B[1<<15],*S=B,*T=B;
#define getc() (S==T&&(T=(S=B)+fread(B,1,1<<15,stdin),S==T)?0:*S++)
#define isd(c) (c>='0'&&c<='9')
int bb;ld aa,ee;ld F(){
    while(ch=getc(),!isd(ch)&&ch!='-');ch=='-'?aa=bb=0:(aa=ch-'0',bb=1);
    while(ch=getc(),isd(ch))aa=aa*10+ch-'0';ee=1;
    if(ch=='.')while(ch=getc(),isd(ch))aa+=(ch-'0')*(ee*=0.1);return bb?aa:-aa;
}
ld G(){
    return (rand()-(1<<30))/1e21;
}
int n,i,j,t[2],vis[1010][1010],now,las,tmp;ld ans;
struct P{ld x,y,z;}p[1010];
#define PP const P&
P operator-(PP a,PP b){return (P){a.x-b.x,a.y-b.y,a.z-b.z};}
P operator&(PP a,PP b){return (P){a.y*b.z-a.z*b.y,a.z*b.x-a.x*b.z,a.x*b.y-a.y*b.x};}
ld operator|(PP a,PP b){return a.x*b.x+a.y*b.y+a.z*b.z;}
ld len(PP a){return sqrt(a.x*a.x+a.y*a.y+a.z*a.z);}
#define ck(a,b,c) ((b-a)&(c-a))
struct Sfc{
    int a,b,c;P s;
```

```
27      void up(int x,int y,int z){
28          a=x,b=y,c=z,s=ck(p[x],p[y],p[z]);
29      }
30  }q[2][3010];
31  #define see(i,f) ((f.s|(p[i]-p[f.a]))>0)
32  #define ns q[las][j]
33  #define pd(a,b) if(vis[a][b]&&!vis[b][a])q[now][++t[now]].up(a,b,i)
34  int main(){
35      scanf("%d",&n);
36      for(i=1;i<=n;i++)p[i]=(P){F()+G(),F()+G(),F()+G()};
37      for(q[1][++t[1]].up(1,2,3),q[1][++t[1]].up(3,2,1),i=4;i<=n;i++){
38          now=i&1,las=now^1,t[now]=0;
39          for(j=1;j<=t[las];j++){
40              if(tmp=see(i,ns),!tmp)q[now][++t[now]]=ns;
41              vis[ns.a][ns.b]=vis[ns.b][ns.c]=vis[ns.c][ns.a]=tmp;
42          }
43          for(j=1;j<=t[las];j++){
44              pd(ns.a,ns.b);
45              pd(ns.b,ns.c);
46              pd(ns.c,ns.a);
47          }
48      }
49      for(now=n&1,i=1;i<=t[now];i++)ans+=len(q[now][i].s);
50      printf("%lf\n",ans*0.5);
51  }
```

8.4 半个半平面交

```
1   #define N 50010
2   int n,i,q[N],r;
3   struct P{int x,y,n;}a[N];
4   #define PP const P&
5   bool operator<(PP a,PP b){return a.x<b.x||a.x==b.x&&a.y<b.y;}
6   P operator-(PP a,PP b){return (P){a.x-b.x,a.y-b.y};}
7   long long operator*(PP a,PP b){return 1LL*a.x*b.y-1LL*a.y*b.x;}
8   long long check(PP a,PP b,PP c){return (b-a)*(c-a);}
9   int main(){
10      for(n=F(),i=1;i<=n;i++)a[i]=(P){F(),-F(),i};
11      std::sort(a+1,a+1+n);
12      for(i=1;i<=n;i++)if(a[i].x!=a[i-1].x){
13          while(r>1&&check(a[q[r-1]],a[q[r]],a[i])<=0)r--;q[++r]=i;
14      }
15      for(i=1;i<=r;i++)q[i]=a[q[i]].n;
16      std::sort(q+1,q+1+r);
17      for(i=1;i<=r;i++)printf("%d␣",q[i]);puts("");
18  }
```

8.5 矩形面积并

```
#include<cstdio>
#include<algorithm>
#include<cstring>
#define N 1024
#define ld double
int n,m,i,j,t,cnt[N],tot;ld ans,len[N],L[N],R[N],dx[N],dy[N],x1,x2,y1,y2;
struct D{ld x1,x2,y;int p;}d[N];
bool operator<(const D&i,const D&j){return i.y<j.y||i.y==j.y&&i.p>j.p;}
void mt(int o){
    if(cnt[o])len[o]=R[o]-L[o];
    else len[o]=len[o<<1]+len[o<<1|1];
}
void bt(int o,int l,int r){
    L[o]=dx[l],R[o]=dx[r],len[o]=cnt[o]=0;
    int mid=l+r>>1;
    if(l+1<r)bt(o<<1,l,mid),bt(o<<1|1,mid,r);
    else len[o<<1]=len[o<<1|1]=0,L[o<<1]=R[o<<1]=L[o],L[o<<1|1]=R[o<<1|1]=R[
        o];
}
void upd(int o,ld l,ld r,int p){
    if(l<=L[o]&&R[o]<=r){
        cnt[o]+=p,mt(o);return;
    }
    if(l<R[o<<1])upd(o<<1,l,r,p);
    if(r>L[o<<1|1])upd(o<<1|1,l,r,p);
    mt(o);
}
int main(){
    while(scanf("%d",&n),n){
        for(tot=m=0,i=1;i<=n;i++)
        scanf("%lf%lf%lf%lf",&x1,&y1,&x2,&y2),
        dy[++tot]=x1,d[tot]=(D){x1,x2,y1,1},
        dy[++tot]=x2,d[tot]=(D){x1,x2,y2,-1};
        std::sort(dy+1,dy+1+tot);
        std::sort(d+1,d+1+tot);
        for(i=1,m=0;i<=tot;i++)
        if(dy[i]!=dy[i+1])dx[++m]=dy[i];
        bt(1,1,m);
        for(ans=0,i=1;i<=tot;i++){
            upd(1,d[i].x1,d[i].x2,d[i].p);
            ans+=len[1]*(d[i+1].y-d[i].y);
        }
        printf("Test␣case␣#%d\nTotal␣explored␣area:␣%.2lf\n\n",++t,ans);
    }
}
```

8.6 曼哈顿距离最小生成树

```
#include<cstdio>
#include<cstring>
#include<algorithm>
char B[1<<15],*S=B,*T=B,ch;
#define getc() (S==T&&(T=(S=B)+fread(B,1,1<<15,stdin),S==T)?0:*S++)
int aa,bb;int F(){
    while(ch=getc(),(ch<'0'||ch>'9')&&ch!='-');ch=='-'?aa=bb=0:(aa=ch-'0',bb
        =1);
    while(ch=getc(),ch>='0'&&ch<='9')aa=aa*10+ch-'0';return bb?aa:-aa;
}
#define N 100010
int n,swp,cnt,z[N];long long ans;
#define swap(a,b) (swp=a,a=b,b=swp)
#define abs(x) (x>0?x:-(x))
#define max(a,b) (a>b?a:b)
#define cmax(x) (ans<x?ans=x:1)
struct P{int x,y,id,nx,ny;}p[N];
bool operator<(const P&a,const P&b){return a.nx<b.nx||a.nx==b.nx&&a.ny<b.ny
    ;}
class Graph{
private:
    int et,la[N],ufs[N],tot;
    struct D{
        int x,y,v;
        bool operator<(const D&a)const{return v<a.v;}
    }d[N<<2];
    struct E{int to,v,nxt;}e[N<<1];
    int gf(int x){return ufs[x]==x?x:ufs[x]=gf(ufs[x]);}
    void adde(int x,int y,int v){
        e[++et]=(E){y,v,la[x]},la[x]=et;
        e[++et]=(E){x,v,la[y]},la[y]=et;
    }
public:
    Graph(){et=1;}
    void add(int x,int y,int v){d[++tot]=(D){x,y,v};}
    void make(){
        std::sort(d+1,d+1+tot);
        for(int i=1;i<=n;i++)ufs[i]=i;cnt=n;
        for(int i=1,x,y;i<=tot;i++)
        if((x=gf(d[i].x))!=(y=gf(d[i].y))){
            ufs[x]=y,cnt--,ans+=d[i].v,
            adde(d[i].x,d[i].y,d[i].v);
        }
    }
}G;
struct D{int x,n;}d[N];
bool operator<(const D&a,const D&b){return a.x<b.x;}
#define dis(i,j) (abs(p[i].x-p[j].x)+abs(p[i].y-p[j].y))
```

```
void ins(int i){
    for(int t=p[i].ny;t<=cnt;t+=t&-t)
    if(z[t]==0||p[z[t]].x+p[z[t]].y<p[i].x+p[i].y)z[t]=i;
}
int query(int i){int f=0;
    for(int t=p[i].ny;t>0;t-=t&-t)
    if(z[t]&&(f==0||p[z[t]].x+p[z[t]].y>p[f].x+p[f].y))f=z[t];
    return f;
}
void work(){
    for(int i=1;i<=n;i++)p[i].nx=p[i].x-p[i].y,p[i].ny=p[i].y;
    std::sort(p+1,p+1+n);
    for(int i=1;i<=n;i++)d[i]=(D){p[i].ny,i};
    std::sort(d+1,d+1+n);d[n+1].x=d[n].x;cnt=1;
    for(int i=1;i<=n;i++){
        p[d[i].n].ny=cnt;
        if(d[i].x!=d[i+1].x)cnt++;
    }
    memset(z,0,sizeof(z));
    for(int i=1,j;i<=n;ins(i++))
    if(j=query(i))G.add(p[i].id,p[j].id,dis(i,j));
}
int main(){
    n=F();
    for(int i=1;i<=n;i++)p[i]=(P){F(),F(),i};work();
    for(int i=1;i<=n;i++)swap(p[i].x,p[i].y);work();
    for(int i=1;i<=n;i++)p[i].y=-p[i].y;work();
    for(int i=1;i<=n;i++)swap(p[i].x,p[i].y);work();G.make();
    printf("%lld\n",ans);
}
```

8.7 欧几里德距离最小生成树

```
#include<cstdio>
#include<cstring>
#include<algorithm>
#include<cmath>
#define N 100010
#define sqr(x) ((x)*(x))
#define max(a,b) (a>b?a:b)
int aa;char ch;int F(){
    while(ch=getchar(),ch<'0'||ch>'9');aa=ch-'0';
    while(ch=getchar(),ch>='0'&&ch<='9')aa=aa*10+ch-'0';return aa;
}
typedef double ld;
struct P{
    ld x,y;
#define PP const P&
    bool operator<(PP a)const {return x<a.x||x==a.x&&y<a.y;}
```

```
17      P operator-(PP a)const {return (P){x-a.x,y-a.y};}
18      ld operator&(PP a)const {return x*a.y-y*a.x;}
19      ld operator|(PP a)const {return x*a.x+y*a.y;}
20  }p[N];
21  #define check(a,b,c) ((b-a)&(c-a))
22  ld dis2(PP a){return a.x*a.x+a.y*a.y;}
23  #define cross(a,b,c,d) (check(p[a],p[c],p[d])*check(p[b],p[c],p[d])<0&&check
        (p[c],p[a],p[b])*check(p[d],p[a],p[b])<0)
24  struct P3{
25      ld x,y,z;
26      bool operator<(const P3&a)const {return x<a.x||x==a.x&&y<a.y;}
27      P3 operator-(const P3&a)const {return (P3){x-a.x,y-a.y,z-a.z};}
28      ld operator|(const P3&a)const {return x*a.x+y*a.y+z*a.z;}
29      P3 operator&(const P3&a)const {return (P3){y*a.z-z*a.y,z*a.x-x*a.z,x*a.y
            -y*a.x};}
30  }ori[N];
31  #define gp3(a) (P3){a.x,a.y,a.x*a.x+a.y*a.y}
32  bool incir(int a,int b,int c,int d){
33      P3 aa=gp3(p[a]),bb=gp3(p[b]),cc=gp3(p[c]),dd=gp3(p[d]);
34      if(check(p[a],p[b],p[c])<0)std::swap(bb,cc);
35      return (check(aa,bb,cc)|(dd-aa))<0;
36  }
37  int n,m,i,j,et=1,la[N],ts,xx,yy,fa[N][20],tot,cnt,dep[N],l,r,q[N<<2],ufs[N];
        ld mx[N][20];
38  struct E{int to,l,r;}e[N<<5];
39  void add(int x,int y){
40      e[++et]=(E){y,la[x]},e[la[x]].r=et,la[x]=et;
41      e[++et]=(E){x,la[y]},e[la[y]].r=et,la[y]=et;
42  }
43  void del(int x){
44      e[e[x].r].l=e[x].l,e[e[x].l].r=e[x].r,la[e[x^1].to]==x?la[e[x^1].to]=e[x
            ].l:1;
45  }
46  void delaunay(int l,int r){
47      if(r-l<=2){
48          for(int i=l;i<r;i++)
49          for(int j=i+1;j<=r;j++)add(i,j);
50          return;
51      }
52      int i,j,mid=l+r>>1,ld=0,rd=0,id,op;
53      delaunay(l,mid),delaunay(mid+1,r);
54      for(tot=0,i=l;i<=r;q[++tot]=i++)
55      while(tot>1&&check(p[q[tot-1]],p[q[tot]],p[i])<0)tot--;
56      for(i=1;i<tot&&!ld;i++)if(q[i]<=mid&&mid<q[i+1])ld=q[i],rd=q[i+1];
57      for(;add(ld,rd),1;){
58          id=op=0;
59          for(i=la[ld];i;i=e[i].l)if(check(p[ld],p[rd],p[e[i].to])>0)
60          if(!id||incir(ld,rd,id,e[i].to))op=-1,id=e[i].to;
61          for(i=la[rd];i;i=e[i].l)if(check(p[rd],p[ld],p[e[i].to])<0)
```

```
62          if(!id||incir(ld,rd,id,e[i].to))op=1,id=e[i].to;
63          if(op==0)break;
64          if(op==-1){
65              for(i=la[ld];i;i=e[i].l)
66              if(cross(rd,id,ld,e[i].to))del(i),del(i^1),i=e[i].r;
67              ld=id;
68          }else{
69              for(i=la[rd];i;i=e[i].l)
70              if(cross(ld,id,rd,e[i].to))del(i),del(i^1),i=e[i].r;
71              rd=id;
72          }
73      }
74  }
75  struct D{int x,y;ld v;}d[N<<3];
76  bool operator<(const D&i,const D&j){return i.v<j.v;}
77  int gf(int x){return ufs[x]==x?x:ufs[x]=gf(ufs[x]);}
78  struct G{int to;double v;int nxt;}g[N<<3];
79  #define addg(x,y,v) (g[++et]=(G){y,v,la[x]},la[x]=et)
80  ld query(int x,int y){
81      int k,i;ld ans=0;
82      if(dep[x]<dep[y])k=x,x=y,y=k;
83      for(k=dep[x]-dep[y],i=0;k;k>>=1,i++)if(k&1)
84      ans=max(ans,mx[x][i]),x=fa[x][i];
85      if(x==y)return ans;
86
87      for(i=0;fa[x][i]!=fa[y][i];i++);
88      for(i--;~i;i--)if(fa[x][i]!=fa[y][i])
89      ans=max(ans,max(mx[x][i],mx[y][i])),x=fa[x][i],y=fa[y][i];
90      ans=max(ans,max(mx[x][0],mx[y][0]));return ans;
91  }
92  int main(){
93      for(n=F(),i=1;i<=n;i++)xx=F(),yy=F(),p[i]=(P){xx,yy},ori[i]=(P3){xx,yy,i
            },ufs[i]=i;
94      std::sort(p+1,p+1+n);std::sort(ori+1,ori+1+n);delaunay(1,n);
95
96      for(i=1;i<=n;i++)
97      for(j=la[i];j;j=e[j].l)xx=ori[i].z,yy=ori[e[j].to].z,
98      d[++tot]=(D){xx,yy,dis2(p[i]-p[e[j].to])};
99      std::sort(d+1,d+1+tot);
100
101     memset(la,0,sizeof(la)),et=0;
102     for(i=1;i<=tot&&cnt<n-1;i++)if(gf(d[i].x)!=gf(d[i].y))
103     cnt++,ufs[ufs[d[i].x]]=ufs[d[i].y],
104     addg(d[i].x,d[i].y,d[i].v),addg(d[i].y,d[i].x,d[i].v);
105
106     for(q[l=r=1]=dep[1]=1;l<=r;l++)
107     for(i=la[q[l]];i;i=g[i].nxt)if(!dep[g[i].to])
108     for(dep[q[++r]=g[i].to]=dep[q[l]]+1,fa[g[i].to][j=0]=q[l],mx[g[i].to
            ][0]=g[i].v;fa[g[i].to][j];j++)
```

```
109     fa[g[i].to][j+1]=fa[fa[g[i].to][j]][j],mx[g[i].to][j+1]=max(mx[g[i].to][
            j],mx[fa[g[i].to][j]][j]);
110
111     for(m=F();m--;printf("%.6lf\n",sqrt(query(F(),F()))));
112 }
```

8.8 平面图点定位

```
1  #include<cstdio>
2  #include<cmath>
3  #include<algorithm>
4  #include<set>
5  int aa;char ch,B[1<<15],*S=B,*T=B;
6  #define getc() (S==T&&(T=(S=B)+fread(B,1,1<<15,stdin),S==T)?0:*S++)
7  #define GetAA() \
8      while(ch=getc(),ch<'0'||ch>'9');aa=ch-'0';\
9      while(ch=getc(),ch>='0'&&ch<='9')aa=aa*10+ch-'0'
10 int F(){GetAA();return aa;}
11 int Fl(){GetAA();return aa<<1|(ch=='.'?getc(),1:0);}
12 #define min(a,b) (a<b?a:b)
13 #define max(a,b) (a>b?a:b)
14 #define cmax(a,b) (a<b?a=b:1)
15 typedef double ld;
16 typedef long long ll;
17 #define N 300010
18 int n,m,qtot,et=1,la[N],id[N],cnt,vis[N],inf;ll sum[N];
19 struct E{int to,v,nxt,pre;}e[N];
20 void adde(int x,int y,int v){
21     e[++et]=(E){y,v,la[x]},la[x]=et;
22     e[++et]=(E){x,v,la[y]},la[y]=et;
23 }
24 struct P{int x,y;}p[N];
25 #define PP const P&
26 bool operator<(PP a,PP b){return a.x<b.x||a.x==b.x&&a.y<b.y;}
27 bool operator==(PP a,PP b){return a.x==b.x&&a.y==b.y;}
28 P operator-(PP a,PP b){return (P){a.x-b.x,a.y-b.y};}
29 ll operator*(PP a,PP b){return 1LL*a.x*b.y-1LL*a.y*b.x;}
30 struct D{P to;int n;};
31 bool operator<(const D&a,const D&b){return atan2(a.to.y,a.to.x)<atan2(b.to.y
       ,b.to.x);}
32 struct Q{P p;int n;}a[N];
33 bool operator<(const Q&a,const Q&b){return a.p<b.p||a.p==b.p&&a.n<b.n;}
34 struct Graph{
35     int et,la[N],q[N],l,r,fa[N][20],mx[N][20],vis[N],dep[N],ufs[N],tot;
36     struct E{int to,v,nxt;}e[N<<1];
37     struct H{
38         int x,y,v;
39         bool operator<(const H&b)const{return v<b.v;}
40     }d[N];
```

```
41      int gf(int x){return ufs[x]==x?x:ufs[x]=gf(ufs[x]);}
42      void adde(int x,int y,int v){
43          e[++et]=(E){y,v,la[x]},la[x]=et;
44          e[++et]=(E){x,v,la[y]},la[y]=et;
45      }
46      void add(int x,int y,int v){d[++tot]=(H){x,y,v};}
47      void bfs(){
48          for(q[l=r=0]=inf==1?2:1,vis[inf==1?2:1]=1;l<=r;l++)
49          for(int i=la[q[l]];i;i=e[i].nxt)
50          if(!vis[e[i].to]){
51              vis[q[++r]=e[i].to]=1;mx[e[i].to][0]=e[i].v;
52              dep[e[i].to]=dep[fa[e[i].to][0]=q[l]]+1;
53              for(int j=0;fa[e[i].to][j];j++)
54              fa[e[i].to][j+1]=fa[fa[e[i].to][j]][j],
55              mx[e[i].to][j+1]=max(mx[e[i].to][j],mx[fa[e[i].to][j]][j]);
56          }
57      }
58      int query(int x,int y){
59          int k,i,ans=0;
60          if(x==inf||y==inf||gf(x)!=gf(y))return -1;
61          if(dep[x]<dep[y])k=x,x=y,y=k;
62          for(i=0,k=dep[x]-dep[y];k;k>>=1,i++)if(k&1)cmax(ans,mx[x][i]),x=fa[x
                ][i];
63          if(x==y)return ans;
64          for(i=0;fa[x][i]!=fa[y][i];i++);
65          for(i--;~i;i--)if(fa[x][i]!=fa[y][i])
66          cmax(ans,mx[x][i]),x=fa[x][i],
67          cmax(ans,mx[y][i]),y=fa[y][i];
68          cmax(ans,mx[x][0]),cmax(ans,mx[y][0]);return ans;
69      }
70      void work1(){
71          for(int i=1;i<=cnt;i++)ufs[i]=i;
72          std::sort(d+1,d+1+tot);et=1;
73          for(int i=1,x,y;i<=tot;i++)
74          if((x=gf(d[i].x))!=(y=gf(d[i].y)))
75          ufs[x]=y,adde(d[i].x,d[i].y,d[i].v);
76          bfs();
77      }
78      void work2(){
79          for(int i=1;i<=qtot;i++)printf("%d\n",query(id[i+n],id[i+n+qtot]));
80      }
81  }G;
82  void sort_edge(int x){
83      int tot=0;static D d[N];
84      for(int i=la[x];i;i=e[i].nxt)d[++tot]=(D){p[e[i].to]-p[x],i};
85      std::sort(d+1,d+1+tot);
86      for(int i=2;i<=tot;i++)e[d[i].n].pre=d[i-1].n;
87      e[d[1].n].pre=d[tot].n;
88  }
```

```
89  void get_area(int col,int i){
90      int o=i;P x=p[e[i^1].to];
91      for(vis[i]=col;(i=e[i^1].pre)!=o;vis[i]=col)
92      sum[col]+=(p[e[i^1].to]-x)*(p[e[i].to]-x);
93      if(sum[col]<0)sum[col]=-sum[col];
94  }
95  void work1(){
96      for(int i=1;i<=n;i++)sort_edge(i);
97      for(int i=2;i<=et;i++)if(!vis[i])get_area(++cnt,i);
98      for(int i=1;i<=cnt;i++)if(sum[i]>sum[inf])inf=i;
99      for(int i=2;i<=et;i++)if(vis[i]!=inf&&vis[i^1]!=inf&&vis[i^1]<vis[i])G.
            add(vis[i],vis[i^1],e[i].v);
100 }
101 ld nowx;
102 struct T{
103     int tot;
104     struct Node{
105         ld k,b,x0;int col;
106         bool operator<(const Node&a)const{return k*(nowx+0.001)+b<a.k*(nowx
                +0.001)+a.b||k*(nowx+0.001)+b==a.k*(nowx+0.001)+a.b&&x0<a.x0;}
107     }tr[N];
108     std::set<Node>s;
109     std::set<Node>::iterator it[N];
110     void init(){
111         for(int i=2;i<=et;i+=2){
112             tr[i>>1].k=1.0*(p[e[i].to].y-p[e[i^1].to].y)/(p[e[i].to].x-p[e[i
                    ^1].to].x);
113             tr[i>>1].b=p[e[i].to].y-tr[i>>1].k*p[e[i].to].x;
114             tr[i>>1].x0=min(p[e[i].to].x,p[e[i^1].to].x);
115             if(p[e[i].to].x>p[e[i^1].to].x)tr[i>>1].col=vis[i];else tr[i
                    >>1].col=vis[i^1];
116         }
117         tr[0].col=inf;tr[0].b=-1;
118         tr[tot=(et>>1)+1].k=0,tr[tot].b=1<<30,tr[tot].col=inf;nowx=0;
119         it[0]=s.insert(tr[0]).first;
120         it[tot]=s.insert(tr[tot]).first;tot++;
121     }
122     void ins(int x,int k){nowx=x;it[k]=s.insert(tr[k]).first;}
123     void del(int k){s.erase(it[k]);}
124     void query(int o,int x,int y){
125         nowx=x,tr[tot].k=0,tr[tot].b=y;
126         id[o]=(*--s.lower_bound(tr[tot])).col;
127     }
128 }tree;
129 void work2(){
130     qtot=F();int tot=n;
131     for(int i=1;i<=qtot;i++)id[i]=inf,
132     a[++tot].p=(P){Fl(),Fl()},a[tot].n=i+n,
133     a[++tot].p=(P){Fl(),Fl()},a[tot].n=i+n+qtot;
```

```
    std::sort(a+1,a+1+n+qtot*2);tree.init();
    for(int i=1;i<=tot;i++)
    if(a[i].n<=n){
        for(int j=la[a[i].n];j;j=e[j].nxt)
        if(p[a[i].n].x!=p[e[j].to].x)
        if(p[a[i].n].x>p[e[j].to].x)tree.del(j>>1);
        else tree.ins(p[a[i].n].x,j>>1);
    }
    else tree.query(a[i].n,a[i].p.x,a[i].p.y);
}
int main(){
    n=F(),m=F();G.et=1;
    for(int i=1;i<=n;i++)a[i].p=p[i]=(P){Fl(),Fl()},a[i].n=i;
    for(int i=1,x,y,v;i<=m;i++)x=F(),y=F(),v=F(),adde(x,y,v);
    work1();G.work1();
    work2();G.work2();
}
```

8.9 辛普森积分

```
#include<cstdio>
#include<cmath>
#include<cstring>
#include<algorithm>
#define eps 1e-8
#define ld double
int n,i,ll,rr,t;ld ans;
struct qj{ld l,r;}q[1010];
struct data{int x,y,r,lx,rx;}a[1010];
ld abs(ld a){return a>0?a:-a;}
ld max(ld a,ld b){return a>b?a:b;}
bool cmp(const qj&i,const qj&j){return i.l<j.l;}
ld sqr(ld x){return x*x;}
ld f(ld xx)
{
    int i;ld ans=0,nl=-10001,nr=-10001;t=0;
    for(i=1;i<=n;i++)
    if(abs(a[i].x-xx)<=a[i].r)
    {
        q[++t].l=a[i].y-sqrt(sqr(a[i].r)-sqr(a[i].x-xx));
        q[t].r=2*a[i].y-q[t].l;
    }
    std::sort(q+1,q+1+t,cmp);
    for(i=1;i<=t;i++)
    {
        if(nr<q[i].l){ans+=nr-nl;nl=q[i].l;}
        nr=max(nr,q[i].r);
    }
    return ans+nr-nl;
```

```
}
ld d(ld l,ld r,ld fl,ld fr,ld fm,ld s)
{
    ld mid=(l+r)/2;
    ld flm=f((l+mid)/2);
    ld fmr=f((mid+r)/2);
    ld ls=(fl+fm+4*flm)*(mid-l);
    ld rs=(fm+fr+4*fmr)*(r-mid);
    if(r-l>3.75)return d(l,mid,fl,fm,flm,ls)+d(mid,r,fm,fr,fmr,rs);
    if(abs(s-ls-rs)<eps)return s;
    else return d(l,mid,fl,fm,flm,ls)+d(mid,r,fm,fr,fmr,rs);
}
bool cmp1(const data&i,const data&j){return i.lx<j.lx;}
int main()
{
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        scanf("%d%d%d",&a[i].x,&a[i].y,&a[i].r);
        a[i].lx=a[i].x-a[i].r;
        a[i].rx=a[i].x+a[i].r;
    }
    std::sort(a+1,a+1+n,cmp1);
    ll=rr=-10001;
    ld fl,fr,fm,fs;
    for(i=1;i<=n;i++)
    {
        if(rr<a[i].lx)
        {
            fl=f(ll);fr=f(rr);fm=f((ll+rr)>>1);fs=(fl+fr+4*fm)*(rr-ll);
            ans+=d(ll,rr,fl,fr,fm,fs);
            ll=a[i].lx;
        }
        rr=max(rr,a[i].rx);
    }
    fl=f(ll);fr=f(rr);fm=f((ll+rr)>>1);fs=(fl+fr+4*fm)*(rr-ll);
    ans+=d(ll,rr,fl,fr,fm,fs);
    printf("%.3lf",ans/6);
}
```

8.10 面积并-格林公式

$\iint_D \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} = \oint_C P dx + Q dy$, $let\ \ P = -y,\ \ Q = x$, $\iint_D 2 dx dy = \oint_C x dy - y dx$

```
#include<cstdio>
#include<cmath>
#include<algorithm>
typedef double ld;
const int N = 1010; const ld pi = acos(-1);
int n; ld ans;
```

```
7  struct P {
8      ld x, y;
9      P operator-(const P&a) const {return (P) {x - a.x, y - a.y};}
10     ld len () {return sqrt(x * x + y * y);}
11 };
12 struct C {
13     P o; ld r;
14     bool operator < (const C&a) const {
15         if (o.x != a.o.x) return o.x < a.o.x;
16         if (o.y != a.o.y) return o.y < a.o.y;
17         return r < a.r;
18     }
19     bool operator==(const C&a)const{return o.x==a.o.x&&o.y==a.o.y&&r==a.r;}
20     ld oint (ld t1, ld t2) {
21         return r*(r*(t2-t1)+o.x*(sin(t2)-sin(t1))-o.y*(cos(t2)-cos(t1)));
22     }
23 } a[N];
24 struct D {
25     ld x; int c;
26     bool operator < (const D&a) const {return x < a.x;}
27 } pos[N * 2];
28 ld work (int c) {
29     int tot = 0, cnt = 0;
30     for (int i = 1; i <= n; i++)
31     if (i != c) {
32         P d = a[i].o - a[c].o; ld dis = d.len();
33         if (a[c].r <= a[i].r - dis) return 0;
34         if (a[i].r <= a[c].r - dis || a[i].r + a[c].r <= dis) continue;
35         ld g = atan2(d.y, d.x), g0 = acos((dis * dis + a[c].r * a[c].r - a[i
               ].r * a[i].r) / (2 * dis * a[c].r)), l = g - g0, r = g + g0;
36         if (l < -pi) l += pi * 2;
37         if (r >= pi) r -= pi * 2;
38         if (l > r) cnt++;
39         pos[++tot] = (D) {l, 1};
40         pos[++tot] = (D) {r, -1};
41     }
42     pos[0].x = -pi, pos[++tot].x = pi;
43     std::sort(pos + 1, pos + 1 + tot);
44     ld ans = 0;
45     for (int i = 1; i <= tot; cnt += pos[i++].c)
46         if (cnt == 0) ans += a[c].oint(pos[i - 1].x, pos[i].x);
47     return ans;
48 }
49 int main () {
50     scanf("%d", &n);
51     for(int i=1;i<=n;i++)scanf("%lf%lf%lf",&a[i].o.x,&a[i].o.y,&a[i].r);
52     std::sort(a+1,a+1+n); n=std::unique(a+1,a+1+n)-a-1;
53     for(int i=1;i<=n;i++)ans+=work(i);
54     printf("%.3lf\n",ans/2);
```

```
55 }
```