

# 题目

在一个长方形框子里，最多有 $N$  ( $0 \leq N \leq 6$ ) 个相异的点，在任何一个点上放一个很小的油滴，那么这个油滴会一直扩展，直到接触到其他油滴或者框子的边界。必须等一个油滴扩展完毕才能放置下一个油滴。那么应该按照怎样的顺序在这 $N$ 个点上放置油滴，才能使放置完毕后所有油滴占据的总体积最大呢？（不同的油滴不会相互融合）

注：圆的面积公式 $V = \pi r^2$ ，其中 $r$ 为圆的半径。

# 分析

用stl自带的next\_permutation 枚举油滴扩展的顺序（最多720种）  
每确定一种顺序，求一次答案

贪心地将当前点扩展到最大，如果当前点已经被前面的油滴覆盖到，那么这个点不放油滴，画图可以感性认识，这种情况最大，不知道怎么证明...

最后输出答案，四舍五入就是将当前的小数答案+0.5 再向下取整

# 代码

```
1  #include<bits/stdc++.h>
2  #define For(i,a,b) for(int i=(a); i<=(b) ; i++)
3  #define _For(i,a,b) for(int i=(a); i>=(b) ; i--)
4  #define Memset(a,b); memset((a),(b),sizeof((a)));
5  #define Cout(a,b); printf("%d",(a));printf(b);
6  #define Coutc(a,b); printf("%c",(a));printf(b);
7  #define Couts(a,b); printf("%s",(a));printf(b);
8  using namespace std;
9  const int INF = 0x3f3f3f3f;
10 typedef long long LL;typedef unsigned long long ULL;typedef long
double LDB;
11 inline LL CinLL(){LL x=0,f=1;char ch=getchar();while(ch<'0' || ch>'9')
{if(ch=='-')f=-1;ch=getchar();}while(ch>='0' && ch<='9'){x=x*10+ch-
'0';ch=getchar();}return x*f;}
12 inline int Cin(){int x=0,f=1;char ch=getchar();while(!isdigit(ch))
{if(ch=='-')f=-1;ch=getchar();}while(isdigit(ch))x=x*10+ch-
'0',ch=getchar();return f*x;}
13 struct cre{
14     double x,y,r,are;
15     bool use;
```

```

16 }a[10];
17 const double p = acos(-1);
18 const double eps = 1e-8;
19 double xx1,yy1,xx2,yy2;
20 int n;
21 int sx[] = {0,1,2,3,4,5,6};
22 inline double dis(double xa,double ya,double xb,double yb){
23     return sqrt((xb - xa) * (xb - xa) + (yb - ya ) * (yb - ya));
24 }
25 double solve()
26 {
27     For(i,1,n)
28     {
29         int now = sx[i];
30         double rr = 99999999;
31         int flag = 0;
32         For(j,1,i-1)
33             if(dis(a[now].x,a[now].y,a[sx[j]].x,a[sx[j]].y) <= a[sx[j]].r){
34                 a[now].use = false;flag = 1;break;
35             }
36         if(flag)continue;
37         For(c,1,i-1){
38             int j= sx[c];if(a[j].use == false) continue;
39             double kk = dis(a[now].x,a[now].y,a[j].x,a[j].y) - a[j].r;
40             rr = rr < kk ? rr : kk;
41         }
42         rr = rr < (a[now].x - xx1) ? rr : (a[now].x - xx1);
43         rr = rr < (xx2 - a[now].x) ? rr : (xx2 - a[now].x);
44         rr = rr < (a[now].y - yy1) ? rr : (a[now].y - yy1);
45         rr = rr < (yy2 - a[now].y) ? rr : (yy2 - a[now].y);
46         a[now].r = rr;
47         a[now].are = p * a[now].r * a[now].r;
48         a[now].use = true;
49     }
50     double sum = 0;
51     For(i,1,n) sum += a[i].are;
52     return sum;
53 }
54 }
55 int main()
56 {
57     ios::sync_with_stdio(false);
58     cin>>n;
59     cin>>xx1>>yy1>>xx2>>yy2;
60     if(xx1 > xx2){
61         swap(xx1,xx2);swap(yy1,yy2);

```

```
62     }
63     if(yy1 > yy2) swap(yy1,yy2);
64     For(i,1,n) cin>>a[i].x>>a[i].y;
65     double ans = 0;
66     double tim = 1;
67     For(i,2,n) tim*=i;
68     For(i,1,tim)
69     {
70         For(j,1,n) {
71             a[j].are = a[j].r = 0;
72             a[j].use = false;
73         }
74         double k = solve();
75         ans = ans > k ? ans : k;
76         next_permutation(sx+1,sx+n+1);
77     }
78     double res = (xx2 - xx1) * (yy2 - yy1) - ans;
79     int ress = (res + 0.5+eps);
80     cout<<ress;
81 }
```