

# 2020牛客暑期多校训练营（第七场）

[比赛链接](#)

## A - Social Distancing

Solved by qxforever.

### 题目描述

在半径为  $r$  的圆内选  $n$  个整点，使两两距离平方的和最大，输出答案  $\leq 8$ ,  $r \leq 30$ ,  $T \leq 250$

### 解题思路

注意到  $n, r$  的范围很小，输入最多有  $240$  种情况，因此想到打表来解决此题。

首先所选的点一定在圆内整点形成的凸包上，如果不在凸包上，凸包上一定存在一点使答案更优。计算了一下  $r \in [1, 30]$  的凸包顶点数，发现最多为  $36$ 。在这些点中遍历答案即可，对每组  $(n, r)$  最多有  $\binom{36+8-1}{8} = 1.45 \times 10^8$  种选择方案。本地需要  $\sim 1$  分钟可以打完。

注意在凸包上顶点很多的时候，也是有可能两个点重合的。一开始为了效率进行了这样的剪枝，导致 +2。

感觉这里用概率算法并不是很好。

打表代码:

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef pair<int,int> pii;
const int maxn=1e4+23;

struct Point{
    int x,y;
    Point(int x=0,int y=0):x(x),y(y) {}
    bool operator < (const Point &b){
        return x<b.x || (x==b.x&&y<b.y);
    }
};
Point p[maxn],ch[maxn];int cnt;

typedef Point Vector;
Point operator + (Point A,Point B){return Point(A.x+B.x,A.y+B.y);}
Point operator - (Point A,Point B) {return Point(A.x-B.x,A.y-B.y);}
Point operator * (Point A,double B) {return Point(A.x*B,A.y*B);}
```

```
Point operator / (Point A,double B) {return Point(A.x/B,A.y/B);}

int dot(Vector a,Vector b){
    return a.x*b.x+a.y*b.y;
}
int cross(Vector a,Vector b){
    return a.x*b.y-a.y*b.x;
}

int length(Vector a){
    return a.x*a.x+a.y*a.y;
}

int ConvexHull(Point *p,int n,Point *ch){
    sort(p,p+n);
    int m=0;
    for(int i=0;i<n;i++){
        while(m>1&&cross(ch[m-1]-ch[m-2],p[i]-ch[m-2])<=0) m--;
        ch[m++]=p[i];
    }
    int k=m;
    for(int i=n-2;i>=0;i--){
        while(m>k&&cross(ch[m-1]-ch[m-2],p[i]-ch[m-2])<=0) m--;
        ch[m++]=p[i];
    }
    if(n>1) m--;
    return m;
}

int a[maxn],n,r,sz,ans,vis[100];

void dfs(int p,int dep){
    if(dep==n){
        int sum=0;
        for(int i=0;i<n;i++){
            for(int j=0;j<i;j++){
                int x=a[i],y=a[j];
                sum+=length(ch[y]-ch[x]);
            }
        }
        ans=max(ans,sum);
        return ;
    }
    for(int i=p+1;i<sz;i++){
        a[dep]=i;
        dfs(i,dep+1);
    }
}
```

```

void dfs2(int p,int dep){
    if(dep==n){
        int sum=0;
        for(int i=0;i<n;i++){
            for(int j=0;j<i;j++){
                int x=a[i],y=a[j];
                sum+=length(ch[y]-ch[x]);
            }
        }
        ans=max(ans,sum);
        return ;
    }
    for(int i=p;i<sz;i++){
        a[dep]=i;
        dfs2(i,dep+1);
    }
}

void solve(int x,int y){
    n=x,r=y;
    ans=0;
    cnt=0;
    for(int i=0;i<=r;i++){
        for(int j=0;j<=r;j++){
            if(i*i+j*j<=r*r){
                p[cnt++]=Point(i,j);
                p[cnt++]=Point(-i,j);
                p[cnt++]=Point(i,-j);
                p[cnt++]=Point(-i,-j);
            }
        }
    }
    sz=ConvexHull(p,cnt,ch);
    dfs2(0,0);
    printf("ans[%d][%d]=%d;\n",n,r,ans);
}

int main(){
    // freopen("1.out","w",stdout);
    for(int i=1;i<=8;i++){
        for(int j=1;j<=30;j++) solve(i,j);
    }
}

```

## B - Mask Allocation

Solved by qxforever.

## 题目描述

将  $n \times m$  个数分组，使得存在能选出  $n$  组  $m$  个的方案以及  $m$  组  $n$  个的方案，最小化组数，输出字典序最大的方案。

## 解题思路

将  $n, m$  进行类似辗转相除的过程即可保证组数最小。

## D - Fake News

前缀平方和是完全平方数的正整数只有  $1$  和  $24$

## H - Dividing

Solved by nikkukun & qxforever.

## 题目描述

定义 Legeng Tuple 如下，

- $(1, k)$  是
- 如果  $(n, k)$  是，那么  $(n + k, k)$  也是
- 如果  $(n, k)$  是，那么  $(n * k, k)$  也是

给定  $N, K$  问对任意  $1 \leq n \leq N, 1 \leq k \leq K$  一共有多少 Legeng Tuple  $(n, k)$   $N, K \leq 10^{12}$

## 解题思路

分两种情况考虑

- 进行过  $*k$  操作，那么可以表示为  $p * k$
- 没有进行过  $*k$  操作，那么可以表示为  $p * k + 1$

答案是  $\sum_{i=1}^k (\lfloor \frac{n-1}{i} \rfloor + \lfloor \frac{n}{i} \rfloor + 1)$  可以平方分块，也可以暴力算到  $\sqrt{n}$  后面就是一些  $0$  和  $1$ 。

## 赛后总结

**nikkukun**

**qxforever**

**Potassium**

From:

<https://wiki.cvbbacm.com/> - **CVBB ACM Team**

Permanent link:

[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:i\\_dont\\_know\\_png:multi2020-nowcoder-7&rev=1596735896](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:i_dont_know_png:multi2020-nowcoder-7&rev=1596735896) 

Last update: **2020/08/07 01:44**