

疯狂购物



shopping
— tagline here —

请同学写出题目大意
已知什么求什么

给定 n 个数字和最大差值 k ，第 i 个数字为 x_i ，求 x_i 到 x_i+k 之间最多有多少个数字

请同学阅读[数据规模和约定]
识别部分得分点

【数据规模与约定】
 $1 \leq N \leq 1000$ ， $0 \leq K \leq 10000$

算法一：枚举+统计

枚举

统计

题目描述的情况不确定性的根源是什么？

所有物品中价格最低的一件 x_i

依次将每一件物品作为最低价

统计多少个数字在 $[x_i, x_i+k]$ 范围内

算法时间复杂度 $O(n^2)$

题目 $n \leq 1000$ ，不会超时

枚举+统计

```
12 ans=1;  
13 for(int i=0;i<n;i++) {  
14     ll cnt=0;  
15     for(int j=0;j<n;j++)  
16         if(x[i]<=x[j]&& x[j]<=x[i]+k) cnt++;  
17     if(cnt>ans) ans=cnt;  
18 }
```

算法二： 二分查找

算法一的时间复杂度为 $O(n^2)$ ，统计 $[xi, xi+k]$ 范围内数字数量时要枚举全部数字

如何快速求得 $[xi, xi+k]$ 区间内的数字数量？

upper_bound，返回小于等于某个数字的元素数量

lower_bound，返回小于某个数字的元素数量

做减法就可以计算区间内元素数量

算法时间复杂度 $O(n\log n)$

二分查找

```
12  sort(x,x+n);
13  ans=1;
14  for(int i=0;i<n;i++) {
15      ll lb=lower_bound(x,x+n,x[i])-x;
16      ll up=upper_bound(x,x+n,x[i]+k)-x;
17      ll cnt=up-lb;
18      if(cnt>ans) ans=cnt;
19  }
```

太戈编程

2801

虎年

请同学写出题目大意
已知什么求什么

给出N个年份点，在12的倍数点**至多**穿越K次，求从点0穿越，遍历完所有祖先点再回到点0，最少需要经历多少年

请同学阅读[数据规模和约定]
识别部分得分点

【数据规模与约定】

$1 \leq N \leq 65536$

$0 \leq K \leq N$

注意：所有祖先点都不在12的倍数点

输入样例

5 3

101

85

100

46

95

输出样例

36



说明

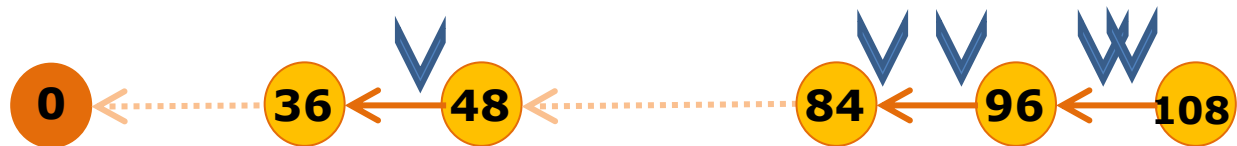
第一次穿越：从0穿越到108

经历两个12年到84

第二次穿越：从84穿越到48

经历12年到36

第三次穿越：从36穿越回到0



输入样例

5 3

101

85

100

46

95

输出样例

36

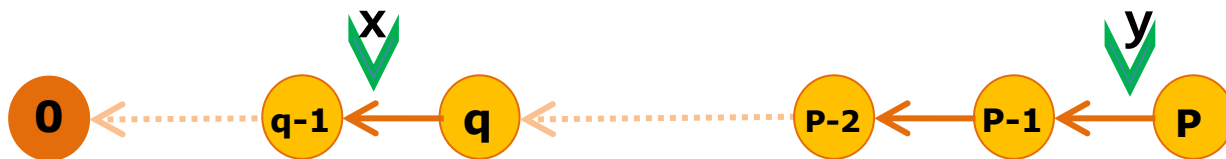
题目分析

12年为一个周期区间

点 a_i 所在区间号为 $\text{ceil}(a_i/12.0)$
若某区间有点，这12年必须经历
否则可以用掉一次穿越

第一次需从0穿越到最远周期点
即 $12 * \text{ceil}(a_{\text{最远点}}/12.0)$

剩下 $k-1$ 次穿越机会
贪心地用给相距较远的两邻居点



实现细节

点直接压缩为区间号 $\text{ceil}(a_i/12.0)$
排序，计算邻居点区间差值，再排序

从相隔最远两点 x 和 y 开始穿越
 x 区间号是 q ， y 区间号是 p
本次穿越可省下 $12*(p-q-1)$ 年

这个过程持续到 $k-1$ 次穿越耗尽
或剩下邻居点都在相邻区间或同区间

注意：源点 0 的区间号是 0
它与最近邻居的区间差也需参与排序！

```
3  const int SIZE=65537;  
4  int n,k,ans,a[SIZE]={0},b[SIZE];
```

```
6      cin>>n>>k;  
7  for(int i=1;i<=n;++i){  
8      cin>>a[i];  
9      a[i]=ceil(a[i]/12.0);  
10 }  
11 sort(a+1,a+n+1);
```

从相隔区间最远的
两邻居点消耗穿越机会
直至 $k-1$ 次耗尽
或剩下邻居点都在相邻
区间或同区间

```
12 ans=a[n]*12;
13 for(int i=0;i<n;++i)
14     b[i]=a[i+1]-a[i];
15 sort(b,b+n);
16 for(int i=n-1;i>=n-(k-1);--i){
17     if(b[i]<=1) break;
18     ans -= ;
19 }
20 cout<<ans;
```

是否需要判下越界?

舒适

请同学写出题目大意
已知什么求什么

对于前*i*只奶牛，求加入最少奶牛数量，
使得不存在在某牛正好三面有邻

请同学阅读[数据规模和约定]
识别部分得分点

【数据规模与约定】

$$1 \leq N \leq 10^5$$

$$0 \leq x_i, y_i \leq 1000$$

注意：为了取消“三面有邻”而需加入的奶牛的*x*和
*y*坐标并不一定需要在范围 0到1000之内

输入样例

9

0 1

1 0

1 1

1 2

2 1

2 2

3 1

3 2

4 1

输出样例

0

0

0

1

0

0

1

2

4

9

8

7

6

5

4

3

2

1

0

0

1

2

3

4

5

6

7

8

9



4号加入后
需在3号上方添加一只牛

输入样例

9

0 1

1 0

1 1

1 2

2 1

2 2

3 1

3 2

4 1

输出样例

0

0

0

1

0

0

1

2

4

9

8

7

6

5

4

3

2

1

0

0

1

2

3

4

5

6

7

8

9

5号占据了虚拟加入的蓝牛1
此时又不再需要额外添加

7号进入后，5号舒适
需添加蓝牛到(2,0)



输入样例

9

0 1

1 0

1 1

1 2

2 1

2 2

3 1

3 2

4 1

输出样例

0

0

0

1

0

0

1

2

4

9

8

7

6

5

4

3

2

1

0

0

1

2

3

4

5

6

7

8

9

8号进入后，6号舒适
需添加蓝牛2号到(2,3)



输入样例

9

0 1

1 0

1 1

1 2

2 1

2 2

3 1

3 2

4 1

输出样例

0

0

0

1

0

0

1

2

4

9

8

7

6

5

4

3

2

1

0

0

1

2

3

4

5

6

7

8

9

9号进入后，7号舒适
需添加蓝牛3号到(3,0)

连锁效应使蓝牛1舒适
需再在(2,-1)添加蓝牛4号



输入样例

9
0 1
1 0
1 1
1 2
2 1
2 2
3 1
3 2
4 1

输出样例

0
0
0
1
0
0
1
2
4

题目细节

红牛不会重叠

所以如果出现重叠
一定是红牛占据先前蓝牛点

蓝牛点不受0到1000范围限制

所以点必须平移

新添加牛可能引发连锁反应

需对牛和牛邻居递归检查

```
4  const int N=5009;  
5  int dx[5]={0,0,0,1,-1};  
6  int dy[5]={0,1,-1,0,0};  
7  int n,x,y,ans=0;  
8  bool cow[N][N];
```

(x,y)既可能
自己三面有邻
也可能帮助四面邻居
变成“三面有邻”

```
22 cin>>n;
23 while(n--){
24     cin>>x>>y;
25     x+=1000;y+=1000;
26
27     else{
28         cow[x][y]=1;
29         for(int i=0;i<=4;i++){
30             int nx=x+dx[i],ny=y+dy[i];
31             dfs(nx,ny);
32         }
33     }
34     cout<<ans<<endl;
35 }
```

若 (x,y) 是三面有邻的牛
则 (tx,ty) 是需添加的蓝牛点

```
7 void dfs(int x,int y){
8     int cnt=0,tx,ty;
9     if(!cow[x][y]) return;
10    for(int i=1;i<=4;i++){
11        int nx=x+dx[i],ny=y+dy[i];
12        if(cow[nx][ny]) cnt++;
13        else tx=nx,ty=ny;
14    }
15    if(cnt!=3) return;
16    cow[tx][ty]=1;
17    ans++;
18    for(int i=0;i<=4;i++)
19        dfs(tx+dx[i],ty+dy[i]);
20 }
```

(tx,ty) 既可能自己又
成为三面有邻的牛

也可能引发其邻居
成为三面有邻的牛

魔法阵



现场挑战

用纸和笔概括题目大意
已知什么求什么

写出算法步骤和复杂度
或者写出难点

限时5分钟

请同学核对题目大意
不能遗漏核心要点

枚举符合要求的所有魔法阵，统计每个物品出现在四个位置各自的魔法阵数量。

手算样例

30 8
1
24
7
28
5
29
26
24

输出？

枚举所有魔法阵

手算样例

30 8

1

24

7

28

5

29

26

24

4 0 0 0

0 0 1 0

0 2 0 0

0 0 1 1

1 3 0 0

0 0 0 2

0 0 2 2

0 0 1 0

共有5个魔法阵，分别为：

物品1,3,7,6，其魔法值分别为1,7,26,29;

物品1,5,2,7，其魔法值分别为1,5,24,26;

物品1,5,7,4，其魔法值分别为1,5,26,28;

物品1,5,8,7，其魔法值分别为1,5,24,26;

物品5,3,4,6，其魔法值分别为5,7,28,29。

这是一个数学问题

设 $x_d - x_c = x$,

$x_b - x_a = 2(x_d - x_c) = 2x$,

$x_b - x_a < (x_c - x_b)/3 \implies x_c - x_b > 6x$



能得多少分？

简单枚举

依次枚举或计算 x_a 、 x_b 、 x_c 、 x_d 。

```
2 #define M 40009
3 #define N 15009
4 using namespace std;
5 int x[M],cnt[M],a[M],b[M],c[M],d[M];
```

```

8      cin>>n>>m;
9      for(int i=0;i<m;i++){
10         cin>>x[i];
11         cnt[x[i]]++;
12     }

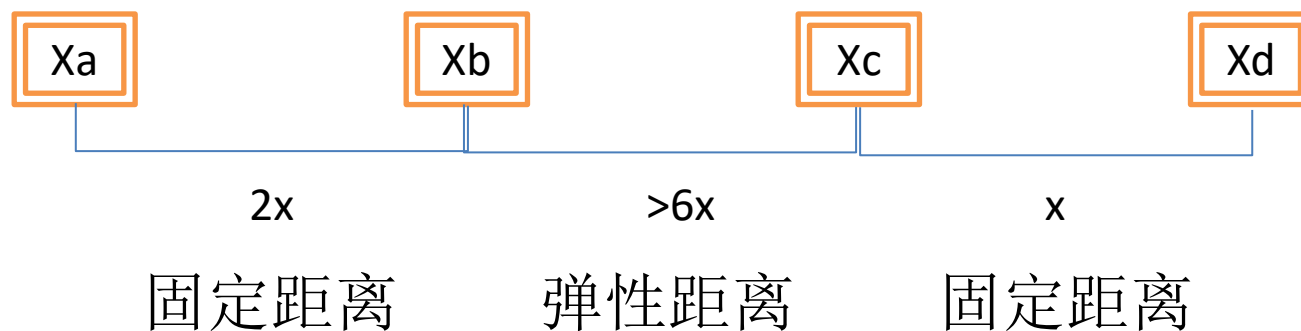
```

太戈编程
www.etiger.vip

简单枚举

```
14  for(int xd=n;xd>=9*i;xd--){           枚举xd
15      if(!cnt[xd])continue;
16      xc=xd-i;
17      if(!cnt[xc])continue;
18      for(int xb=xc-6*i-1;xb>=2*i+1;xb--){
19          if(!cnt[xb])continue;         枚举xb
20          xa=xb-2*i;
21          if(!cnt[xa])continue;
22          a[xa]+=cnt[xb]*cnt[xc]*cnt[xd];
23          b[xb]+=cnt[xa]*cnt[xc]*cnt[xd];
24          c[xc]+=cnt[xa]*cnt[xb]*cnt[xd];
25          d[xd]+=cnt[xa]*cnt[xb]*cnt[xc];
26      }
27 }
```


优化算法



枚举 x ，枚举一端，前缀和计算另一端。

优化算法

从1开始枚举x

```
13 for(int i=1;i<=n/9;i++){
14     int psum=0; // 前缀和
15     for(xd=9*i+2;xd<=n;xd++){
16         xa=xd-9*i-1; xb=xa+2*i; xc=xd-i;
17         psum+=cnt[xa]*cnt[xb];
18         c[xc]+=psum*cnt[xd];
19         d[xd]+=psum*cnt[xc];
20     }
21     psum=0; // 前缀和
22     for(xa=n-9*i-1;xa>=1;xa--){
23         xb=xa+2*i; xc=xb+6*i+1; xd=xc+i;
24         psum+=cnt[xc]*cnt[xd];
25         a[xa]+=psum*cnt[xb];
26         b[xb]+=psum*cnt[xa];
27     }
28 }
```

枚举xd

例如:

xd=11时,
Xa=1, xb=3,
psum+=cnt[1]*cnt[3]

xd=12时,
Xa=2, xb=4,
psum+=cnt[2]*cnt[4]
因为此时xa, xb组合
既可能是1, 3
也可能是2, 4

枚举固定的一端,
弹性的另一端
由前缀和加速计算



```

57 int main(){
58     freopen("magic.in","r",stdin);
59     freopen("magic.out","w",stdout);
60     input();
61     if(n<=1000)
62         solveBF();
63     else
64         solve();
65     return 0;
66 }
```



如何对拍?

保留第62,64行

注释掉第61,63行

正解没写完怎么办?

注释掉61,63,64
保留62

哪怕正解有错
其他部分可以得分