

c6 B小水獭和三点共线

题目描述

小水獭正在补习计算几何。

二维平面上有 n 个不同的点，小水獭请你帮它判断是否存在三点不共线？

输入格式

第一行一个正整数 t ($1 \leq t \leq 10$)，表示数据组数。

对于每组数据，第一行一个正整数 n ($3 \leq n \leq 105$)，含义同题目描述。

接下来 nn 行，每行两个整数 x_i, y_i ($-109 \leq x_i, y_i \leq 109$)，表示第 i 个点的坐标。

保证任意两点坐标不同。

输出格式

对于每组数据，输出一行一个字符串：

- 如果存在三点不共线，输出 `how?`；
- 否则，输出 `boo how! boo how!`。

思路

求三点共线，我们可以用第一个点做起点，利用有公共点的两条直线斜率相等得出这两条直线重合，得出三点共线。即

$$\frac{y_1 - y_2}{x_1 - x_2} = \frac{y_1 - y_3}{x_1 - x_3}$$

但是使用除法会有精度影响。所以可以用乘法

$$(x_1 - x_2)(y_1 - y_3) = (y_1 - y_2)(x_1 - x_3)$$

这样一来只要从1到n循环依次对每个点都讨论剩余所有点，就可以依次求出平面上所有的点是否共线。可是经过实测，这样会tle 😊 因为三层for循环时间复杂度太高了。

然后我们再看看题目，要求是判断是否存在有三点不共线，也就是是否存在两条不重合的直线。那么我们可以直接把第一个点(x_1, y_1)和第二个点(x_2, y_2)作为第一条直线，(x_1, y_1)与剩余的所有点的直线作为第二条直线。这样就只有一层循环，时间复杂度降到n，AC! 😊

代码

```
#include<cstdio>
#include<cmath>
#include<cstdlib>
#include<cstring>
#define ll long long

ll x[200000],y[200000];

bool inlinee(int n){
    int aa = abs(x[1]-x[2]);
```

```

int bb = abs(y[1]-y[2]);
//为了避免有的点相减之后出现负数，所以使用绝对值
//不写也行，因为如果对于(x1,y1)相减得到负值，在相乘时即可得到正
for(int i = 3;i <= n;i++){
    int cc = abs(x[i]-x[1]);
    int dd = abs(y[i]-y[1]);
    if(aa*dd != bb*cc){
        return false;
    }
}
return true;
}
int main(){
    int t;

    scanf("%d",&t);
    while(t > 0){
        t--;
        memset(x,0,sizeof(x));
        memset(y,0,sizeof(y));
        int n;
        scanf("%d",&n);
        for(int i = 1;i <= n;i++){
            scanf("%lld%lld",&x[i],&y[i]);
        }
        bool res = inlinee(n);
        if(res){
            printf("boo how! boo how!\n");
        }else{
            printf("how?\n");
        }
    }
}

```