



## **SC323I: Algorithm Design**

**Section: 541**

**Present**

**Dr. Thitipong Tanprasert**

---

**By**

**JIAN PAN 5728309**

**YILIN WANG 5638021**

**This report is a part of SC323I subject**

**Semester I / 2017**

# TIMUS JUDGE ONLINE 1052. RABBIT HUNT

TIME LIMIT: 1.0 SECOND  
MEMORY LIMIT: 64 MB

DIFFICULTY: 223

---

JIAN PAN      5728309

YILIN WANG    5638021

## 1052. RABBIT HUNT

---

- Time Limit: 2.0 second
- Memory Limit: 16 MB
- A good hunter kills two rabbits with one shot. Of course, it can be easily done since for any two points we can always draw a line containing the both. But killing three or more rabbits in one shot is much more difficult task. To be the best hunter in the world one should be able to kill the maximal possible number of rabbits.
- Assume that rabbit is a point on the plane with integer x and y coordinates. Having a set of rabbits you are to find the largest number of rabbits that can be killed with single shot, i.e. maximum number of points lying exactly on the same line. No two rabbits sit at one point.
- <http://acm.timus.ru/problem.aspx?space=1&num=1052>

# 1052. RABBIT HUNT

---

- **Input**
- An input contains an integer  $N$  ( $2 \leq N \leq 200$ ) specifying the number of rabbits. Each of the next  $N$  lines in the input contains the  $x$  coordinate and the  $y$  coordinate (in this order) separated by a space ( $-1000 \leq x, y \leq 1000$ ).
- **Output**
- The output contains the maximal number of rabbits situated in one line.

input	output
6 7 122 8 139 9 156 10 173 11 190 -100 1	5

# 1052. RABBIT HUNT

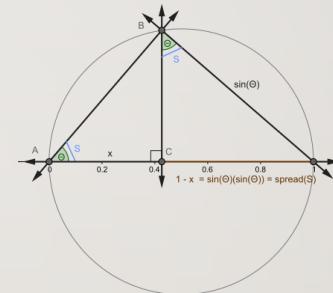
---

The point is that how we can know 3 and more points in one line?

There are so many ways:

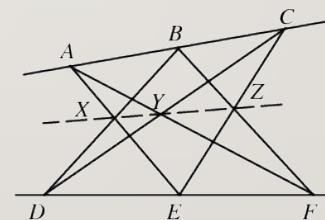
I. Rational trigonometry

$$\sin \angle BAD/AC + \sin \angle CAD/AB = \sin \angle BAC/AD$$



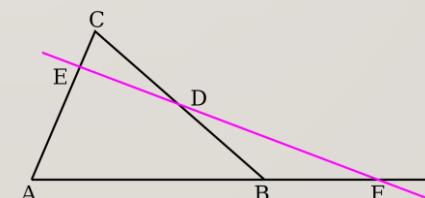
2. Pappus's Hexagon Theorem

<http://mathworld.wolfram.com/PappussHexagonTheorem.html>



3. Menelaus' theorem

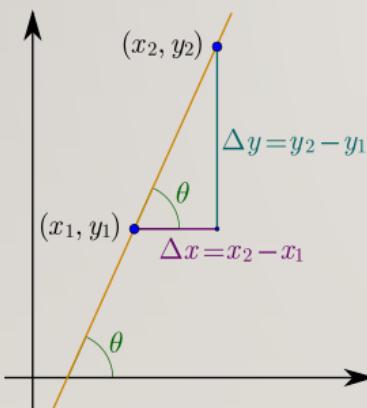
$$\frac{AF}{FB} \times \frac{BD}{DC} \times \frac{CE}{EA} = 1$$



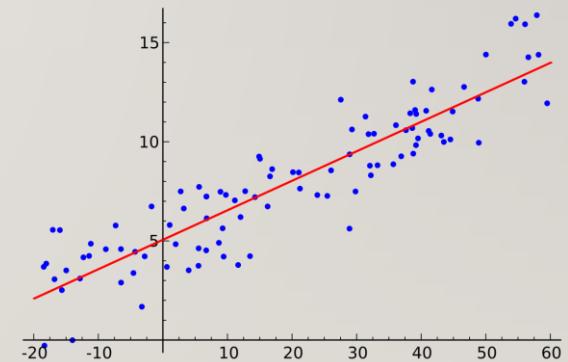
# I052. RABBIT HUNT

---

- However, all of above methods are too complexed
- We just use the most naïve and brute force – calculate slopes ( $O(n^3)$ ).
- An input contains an integer  $N$  ( $3 \leq N \leq 200$ ) specifying the number of rabbits.



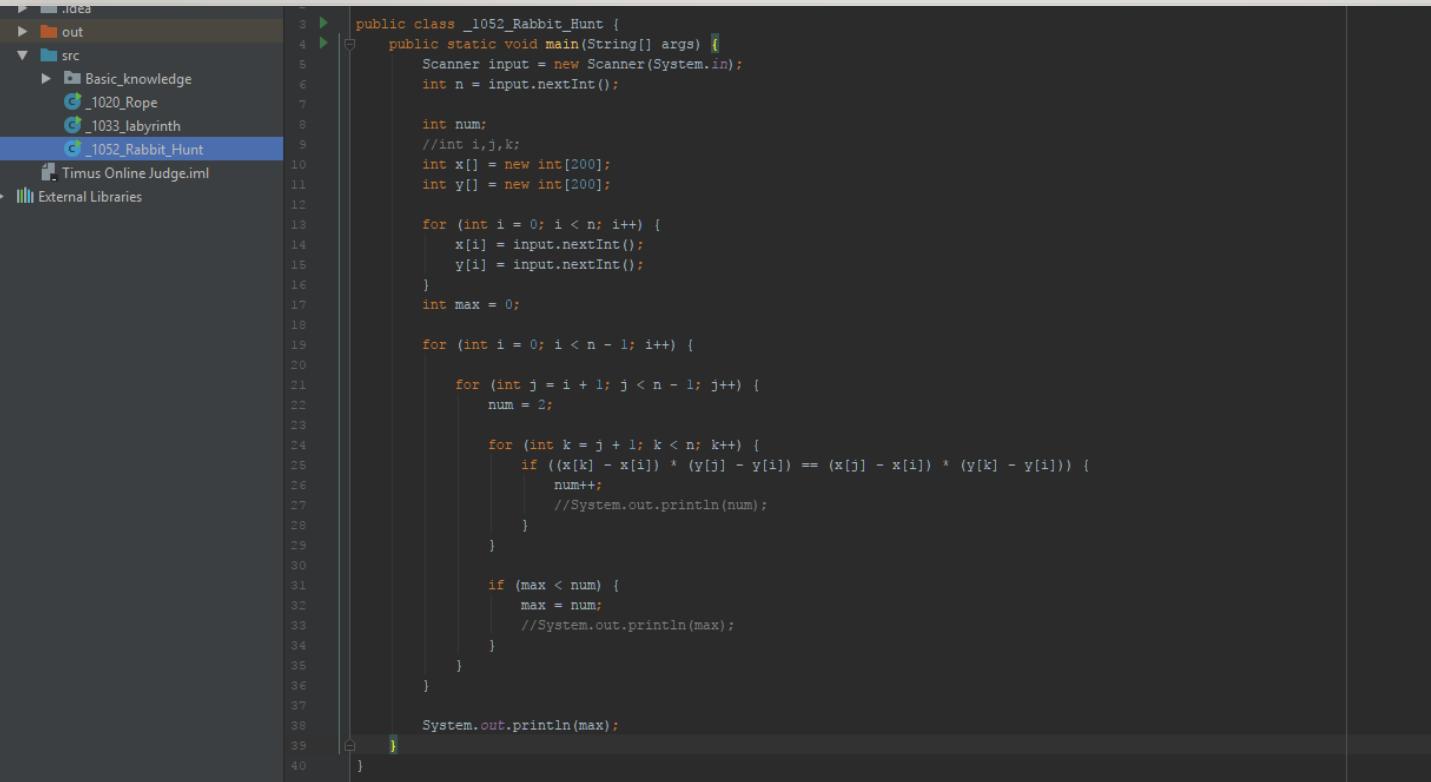
$$(x_3 - x_1)(y_2 - y_1) = (y_3 - y_1)(x_2 - x_1)$$



# 1052. RABBIT HUNT

---

- [https://github.com/jeromepan/Timus-Online-Judge-Solution/blob/master/\\_1052\\_Rabbit\\_Hunt.java](https://github.com/jeromepan/Timus-Online-Judge-Solution/blob/master/_1052_Rabbit_Hunt.java)



The screenshot shows a Java code editor within an IDE. The project structure on the left includes a .idea folder, an out folder, a src folder containing files Basic\_knowledge, \_1020\_Rope, \_1033\_labyrinth, and the selected file \_1052\_Rabbit\_Hunt, and an External Libraries section. The code itself is as follows:

```
public class _1052_Rabbit_Hunt {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int n = input.nextInt();

        int num;
        //int i,j,k;
        int x[] = new int[200];
        int y[] = new int[200];

        for (int i = 0; i < n; i++) {
            x[i] = input.nextInt();
            y[i] = input.nextInt();
        }
        int max = 0;

        for (int i = 0; i < n - 1; i++) {

            for (int j = i + 1; j < n - 1; j++) {
                num = 2;

                for (int k = j + 1; k < n; k++) {
                    if ((x[k] - x[i]) * (y[j] - y[i]) == (x[j] - x[i]) * (y[k] - y[i])) {
                        num++;
                        //System.out.println(num);
                    }
                }

                if (max < num) {
                    max = num;
                    //System.out.println(max);
                }
            }
        }

        System.out.println(max);
    }
}
```

# Result

---

```
<terminated> RabbitHunt [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe
6
7 122
8 139
9 156
10 173
11 190
-100 1
5|
```

# 1052. RABBIT HUNT

---

<a href="#">76292 29</a>	22:44:15 18 Nov 2017	jerrypan	<a href="#">1052. Rabbit Hunt</a>	Java 1.8	Accepted		0.14	2 584 KB
<a href="#">76292 27</a>	22:43:45 18 Nov 2017	jerrypan	<a href="#">1052. Rabbit Hunt</a>	Java 1.8	Wrong answer	I	0.078	2 076 KB
<a href="#">76292 15</a>	22:29:06 18 Nov 2017	jerrypan	<a href="#">1052. Rabbit Hunt</a>	Java 1.8	Wrong answer	2	0.109	2 020 KB
<a href="#">76292 14</a>	22:28:22 18 Nov 2017	jerrypan	<a href="#">1052. Rabbit Hunt</a>	Java 1.8	Wrong answer	I	0.062	2 136 KB
<a href="#">76292 00</a>	22:15:41 18 Nov 2017	jerrypan	<a href="#">1052. Rabbit Hunt</a>	Java 1.8	Wrong answer	2	0.062	2 040 KB
<a href="#">76291 85</a>	22:00:36 18 Nov 2017	jerrypan	<a href="#">1052. Rabbit Hunt</a>	Java 1.8	Runtime error	I	0.078	2 184 KB