Codechef

Ant Colony

Prof. Edson Alves

Faculdade UnB Gama

Codechef – Ant Colony

Problema

In an ants' colony spread over a flat surface, the ants can reside at integral coordinates as close as unit distance apart from each other (but no closer). The colony has the shape of a quadrilateral $ABCD$ where one ant must reside at each of A, B, C, and D, and all other ants can be on any integral coordinate on its perimeter and its interior. Given the integral positions of A, B, C, and D, print the maximum number of ants that can reside in the colony (including the ants that can reside on the four corners and the perimeter).

Entrada e saída

Input

The first line contains the number of test cases, N.

For each test case, a single line contains the x and y coordinates of the four corners of the quadrilateral in clockwise order starting with the left-most, bottom-most point.

Output

For each test case, print the case number, followed by a colon, followed by a single space, followed by a single integer indicating the maximum number of ants.

Constraints

$O < N \leq 3$

$|x|, |y| \leq 10,000$

Exemplo de entradas e saídas

Sample Input

```
2
1 1 3 4 5 3 6 1
1 3 5 6 6 3 3 1
```

Sample Output

```
Case 1: 14
Case 2: 16
```

Solução com complexidade $O(N)$

- Este problema é semelhante ao anterior, com pequenas alterações

Solução com complexidade $O(N)$

- Este problema é semelhante ao anterior, com pequenas alterações
- A primeira delas é que o polígono a ser avaliado é sempre um quadrilátero

Solução com complexidade $O(N)$

- Este problema é semelhante ao anterior, com pequenas alterações
- A primeira delas é que o polígono a ser avaliado é sempre um quadrilátero
- Outro ponto é a orientação do polígono, que passa a ser fixa

Solução com complexidade $O(N)$

- Este problema é semelhante ao anterior, com pequenas alterações
- A primeira delas é que o polígono a ser avaliado é sempre um quadrilátero
- Outro ponto é a orientação do polígono, que passa a ser fixa
- Além disso, a resposta consiste na soma dos pontos interiores I e dos pontos da borda B

Solução com complexidade $O(N)$

- Este problema é semelhante ao anterior, com pequenas alterações
- A primeira delas é que o polígono a ser avaliado é sempre um quadrilátero
- Outro ponto é a orientação do polígono, que passa a ser fixa
- Além disso, a resposta consiste na soma dos pontos interiores I e dos pontos da borda B
- A saída contém a descrição de cada caso de teste

Solução com complexidade $O(N)$

- Este problema é semelhante ao anterior, com pequenas alterações
- A primeira delas é que o polígono a ser avaliado é sempre um quadrilátero
- Outro ponto é a orientação do polígono, que passa a ser fixa
- Além disso, a resposta consiste na soma dos pontos interiores I e dos pontos da borda B
- A saída contém a descrição de cada caso de teste
- Como o número de operações é constante para cada um dos N casos de teste, a solução é $O(N)$

Solução $O(N)$

```cpp
#include <bits/stdc++.h>

using namespace std;
using ll = long long;

struct Point { ll x, y; };

ll gcd(ll a, ll b) { return b ? gcd(b, a % b) : a; }

ll area(int N, const vector<Point>& ps)
{
    ll A = 0;

    for (int i = 0; i < N; ++i)
    {
        A += ps[i].x * ps[i + 1].y;
        A -= ps[i].y * ps[i + 1].x;
    }
```

Solução $O(N)$

```
20      return llabs(A);
21  }
22
23  // Teorema de Pick: A = I + B/2 - 1
24  ll solve(int N, vector<Point>& ps)
25  {
26      ps.push_back(ps.front());
27
28      ll B = 0;
29
30      for (int i = 0; i < N; ++i)
31      {
32          auto b = llabs(ps[i].x - ps[i + 1].x);
33          auto h = llabs(ps[i].y - ps[i + 1].y);
34          auto d = gcd(b, h);
35
36          B += (d + 1);
37      }
```

Solução $O(N)$

```
39      // Desconta os vértices, contados em duplicidade
40      B -= N;
41
42      auto _2A = area(N, ps);
43      auto I = (_2A - B + 2)/2;
44
45      return I + B;
46  }
```