Geometria Computacional

Triângulos: problemas resolvidos

Prof. Edson Alves

2018

Faculdade UnB Gama

Sumário

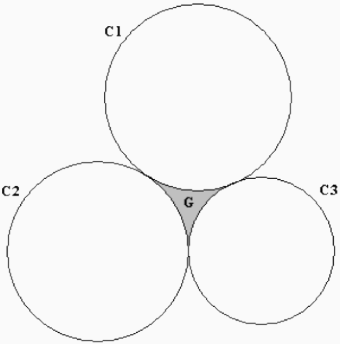
UVA 10991 – Region

Problema

From the figure on the right, it is clear that C1, C2 and C3 circles are touching each other.
Consider,

- C1 circle have R1 radius.
- C2 circle have R2 radius.
- C3 circle have R3 radius.

Write a program that will calculate the area of shaded region G.



Input

The first line will contain an integer k $(1 \leq k \leq 1000)$ which is the number of cases to solve. Each of the following k lines will contain three floating point number $R1$ $(1 \leq R1 \leq 1000)$, $R2$ $(1 \leq R2 \leq 1000)$ and $R3$ $(1 \leq R3 \leq 1000)$.

Output

For each line of input, generate one line of output containing the area of G rounded to six decimal digits after the decimal point. Floating-point errors will be ignored by special judge program.

Exemplo de entradas e saídas

Sample Input

```
2
5.70 1.00 7.89
478.61 759.84 28.36
```

Sample Output

```
1.2243
2361.0058
```

Solução $O(k)$

- A solução parte da observação que os centros dos círculos formam um triângulo T cujos lados são dados por

$$a = R2 + R3, \quad b = R1 + R3, \quad c = R1 + R2$$

- A área do triângulo T contém a região G e mais três setores gerados pelos ângulos opostos a cada um destes lados
- O ângulo oposto a um lado pode ser obtido através da Lei dos Cossenos
- Por exemplo, o ângulo α oposto ao lado a é dado por

$$\alpha = \cos^{-1}\left(\frac{b^2 + c^2 - a^2}{2bc}\right)$$

- A área de cada setor é igual a metade do produto do ângulo pelo quadrado do raio
- Logo a área G é igual a área de T menos a área dos três setores

Solução com complexidade $O(k)$

```cpp
#include <iostream>
#include <cmath>

using namespace std;

const double PI { acos(-1.0) };

double solve(double r1, double r2, double r3)
{
    auto a = r2 + r3;
    auto b = r1 + r3;
    auto c = r1 + r2;
    auto s = (a + b + c)/2.0;
    auto T = sqrt(s*(s - a)*(s - b)*(s - c));

    auto oa = acos((a*a - b*b - c*c)/(-2*b*c));
    auto Sa = 0.5*oa*r1*r1;
    auto ob = acos((b*b - a*a - c*c)/(-2*a*c));
    auto Sb = 0.5*ob*r2*r2;
    auto oc = acos((c*c - b*b - a*a)/(-2*b*a));
    auto Sc = 0.5*oc*r3*r3;
```

Solução com complexidade $O(k)$

```
    auto G = T - Sa - Sb - Sc;

    return G;
}

int main()
{
    int k;
    scanf("%d", &k);

    while (k--)
    {
        double r1, r2, r3;
        scanf("%lf %lf %lf", &r1, &r2, &r3);

        printf("%.6f\n", area);
    }

    return 0;
}
```

Educational Codeforces Round 50 – Problem A: Function Height

Problema

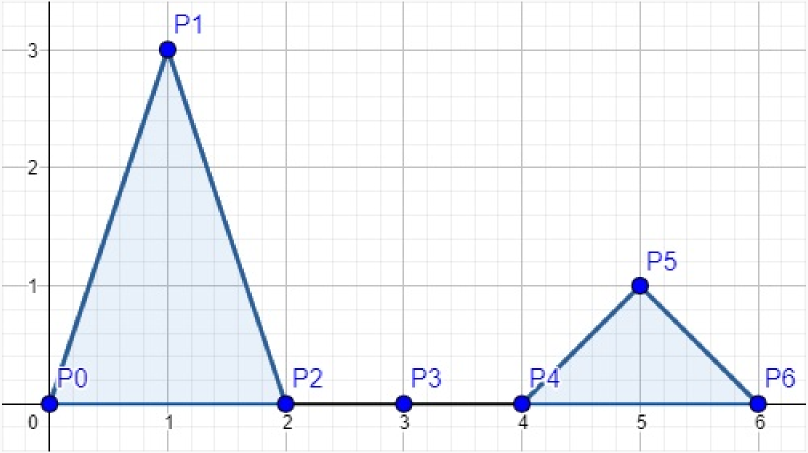
You are given a set of $2n+1$ integer points on a Cartesian plane. Points are numbered from 0 to $2n$ inclusive. Let P_i be the i-th point. The x-coordinate of the point P_i equals i. The y-coordinate of the point P_i equals zero (initially). Thus, initially $P_i = (i, 0)$.

The given points are vertices of a plot of a piecewise function. The j-th piece of the function is the segment $P_j P_{j+1}$.

In one move you can increase the y-coordinate of any point with odd x-coordinate (i.e. such points are $P_1, P_3, \dots, P_{2n-1}$) by 1. Note that the corresponding segments also change.

For example, the following plot shows a function for $n = 3$ (i.e. number of points is $2 \cdot 3 + 1 = 7$) in which we increased the y-coordinate of the point P_1 three times and y-coordinate of the point P_5 one time:

Problema



Problema

Let the area of the plot be the area below this plot and above the coordinate axis OX. For example, the area of the plot on the picture above is 4 (the light blue area on the picture above is the area of the plot drawn on it).

Let the height of the plot be the maximum y-coordinate among all initial points in the plot (i.e. points $P_0, P_1, \ldots, P_{2n}$). The height of the plot on the picture above is 3.

Your problem is to say which minimum possible height can have the plot consisting of $2n+1$ vertices and having an area equal to k. Note that it is unnecessary to minimize the number of moves.

It is easy to see that any answer which can be obtained by performing moves described above always exists and is an integer number not exceeding 10^{18}.

Entrada e saída

Input

The first line of the input contains two integers n and k $(1 \leq n, k \leq 10^{18})$ – the number of vertices in a plot of a piecewise function and the area we need to obtain.

Output

Print one integer – the minimum possible height of a plot consisting of $2n + 1$ vertices and with an area equals k. It is easy to see that any answer which can be obtained by performing moves described above always exists and is an integer number not exceeding 10^{18}.

Exemplo de entradas e saídas

Sample Input	Sample Output
4 3	1
4 12	3
9999999999999999999 9999999999999999986	1

Solução com complexidade $O(1)$

- Não é necessário realizar todas as operações para a solução do problema
- Primeiramente observe que há N triângulos cuja base tem tamanho 2 e altura P_i, com i ímpar
- Ao adicionar uma unidade em P_i a área também aumenta em uma unidade, pois

$$A_{P_i+1} = \frac{2 \times (P_i + 1)}{2} = P_i + 1 = \frac{2 \times P_i}{2} + 1 = A_{P_i} + 1$$

- Se a área total k for distribuída igualmente entre os N triângulos, a altura máxima h a ser atingida é o menor inteiro maior ou igual ao quociente entre N e k, isto é

$$h = \left\lceil \frac{N}{k} \right\rceil$$

Solução AC com complexidade $O(1)$

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

using namespace std;
using ll = long long;

ll solve(ll N, ll K)
{
    return (K + N - 1)/N;
}

int main()
{
    ios::sync_with_stdio(false);

    ll N, K;
    cin >> N >> K;

    cout << solve(N, K) << '\n';

    return 0;
}
```

Referências

1. UVA 10991 – Region
2. CF 1036A – Function Height