

Задача №1075

«Нитка в пространстве»

Кукряков Данил

Условие задачи

Даны координаты 3-х точек:

Точка А — начало

Точка В — конец

Точка С — центр шара

Все значения — целые числа:

$$-1000 \leq (x, y, z) \leq 1000$$

Шар имеет радиус R:

$$\rho(A, C) > R ; \rho(B, C) > R$$

Необходимо протянуть из точки А в точку В нитку минимальной длины.

Задача: Найти длину этой нитки.

Пример работы

Исходные данные:

0 0 12 // координаты точки А

12 0 0 // координаты точки В

10 0 10 // координаты точки С

10 // радиус шара

Результат:

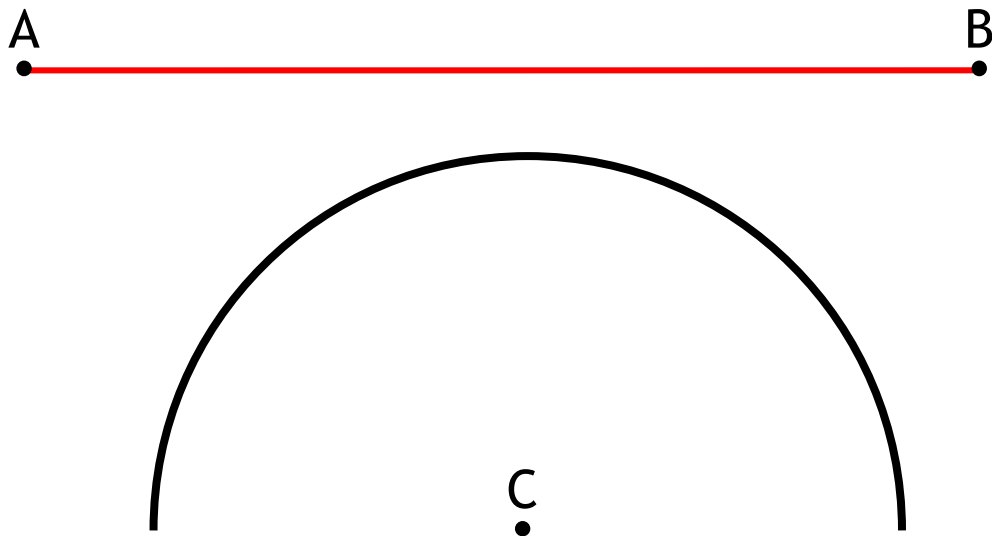
19.71 // длина нитки, округлённая до 2 знаков после точки

Идея алгоритма

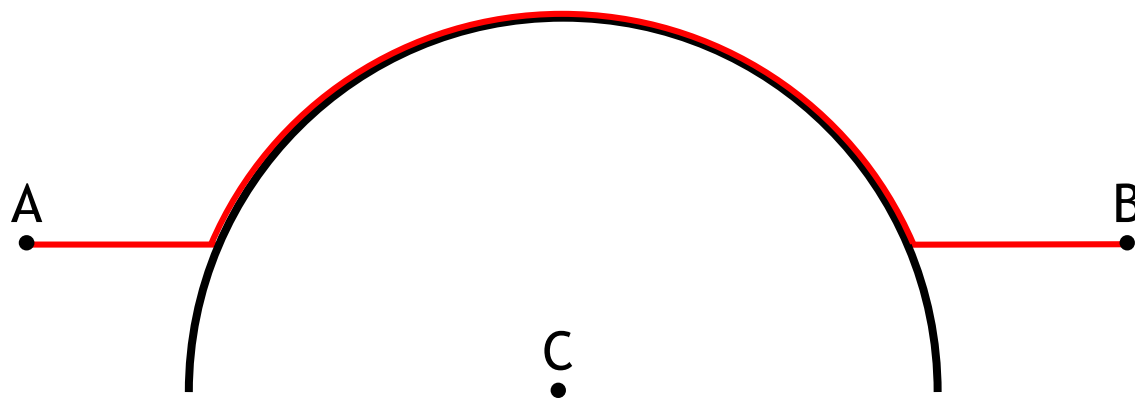
Построим плоскость через точки A, B и C.

Возможны 2 случая:

Нитка идет по прямой



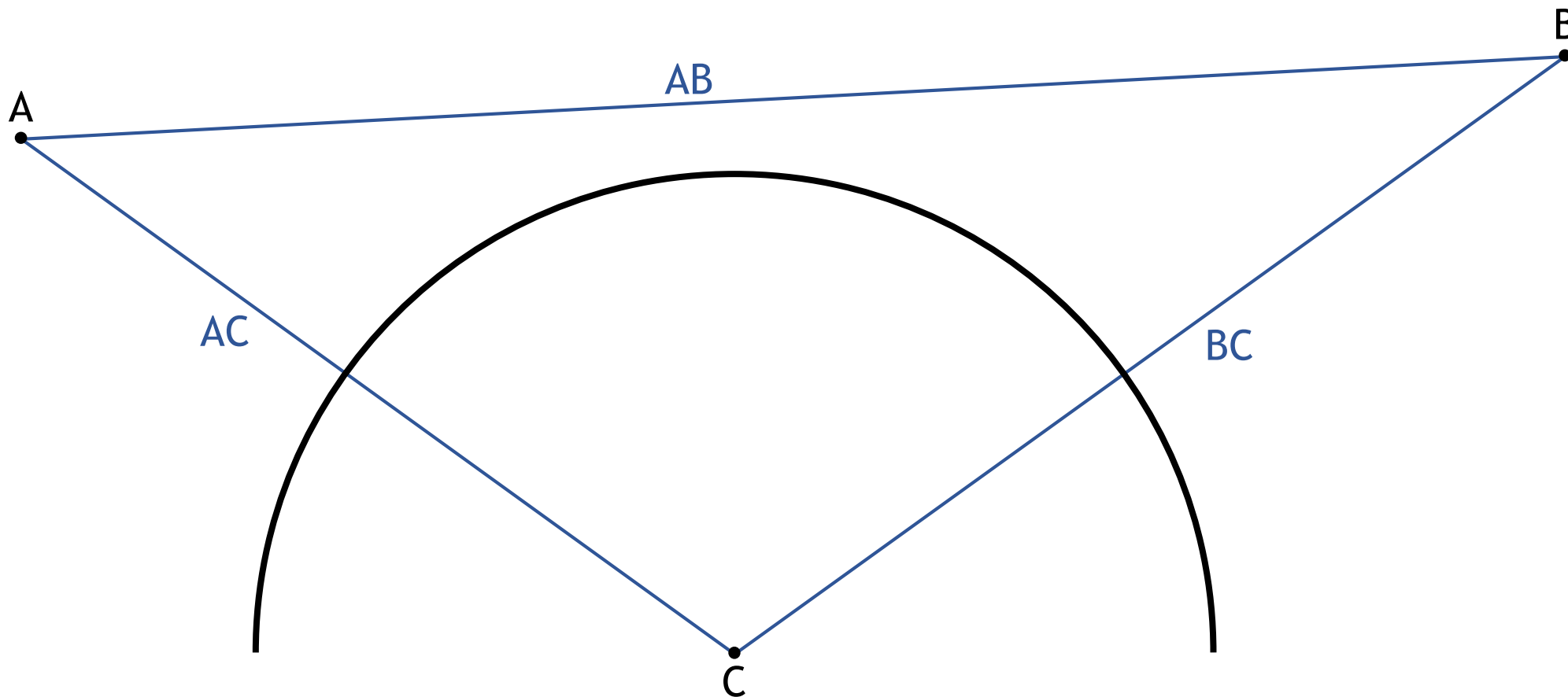
Нитка идет по шару



Нитка идет по прямой

Найдем расстояния АВ, АС и ВС.

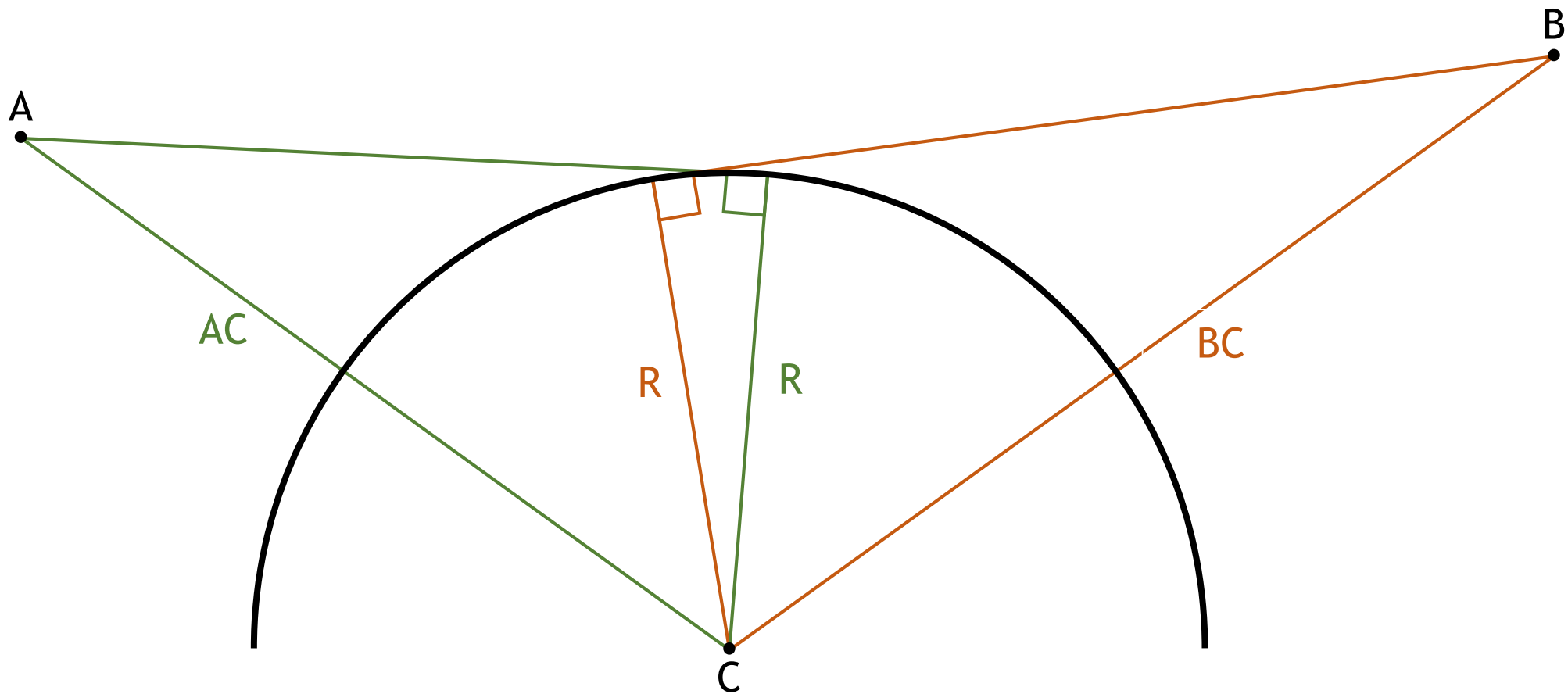
Используем формулу: $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$



Нитка идет по прямой

Построим касательные к окружности из точек A и B.

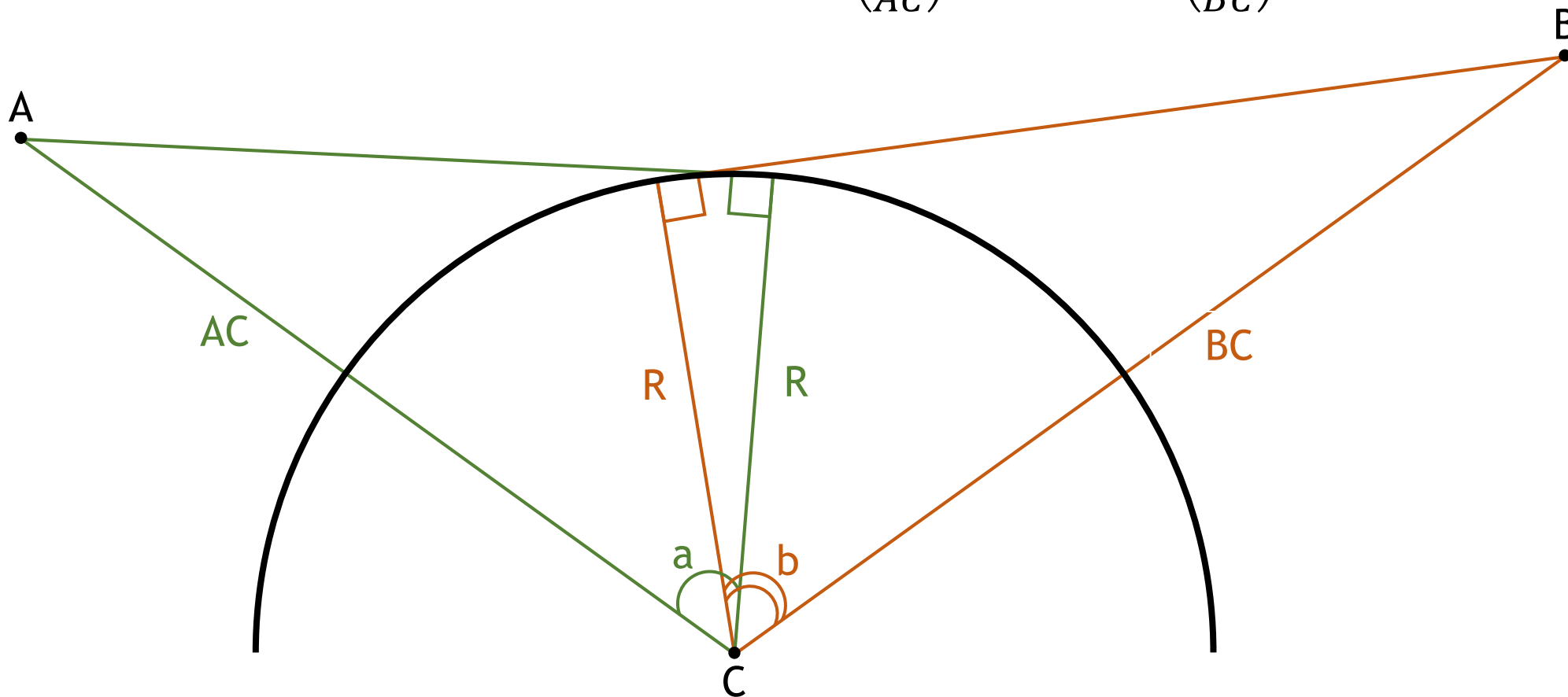
Мы получим два прямоугольных треугольника:



Нитка идет по прямой

В этих треугольниках найдем углы a и b .

Воспользуемся формулами: $a = \arccos\left(\frac{R}{AC}\right)$; $b = \arccos\left(\frac{R}{BC}\right)$;

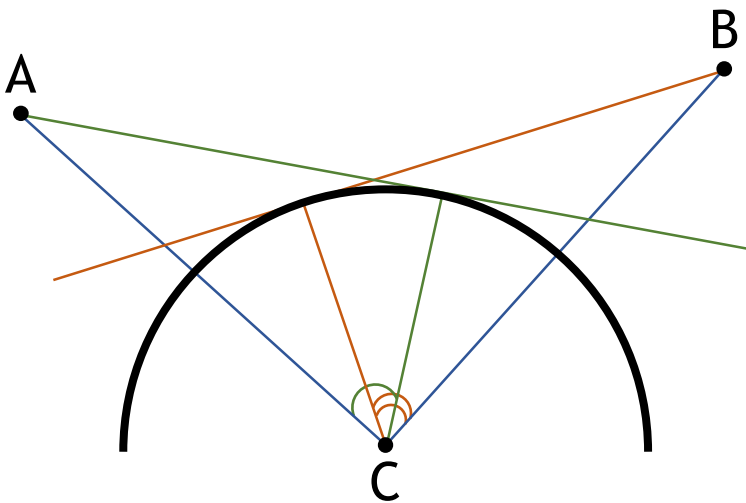


Нитка идет по прямой

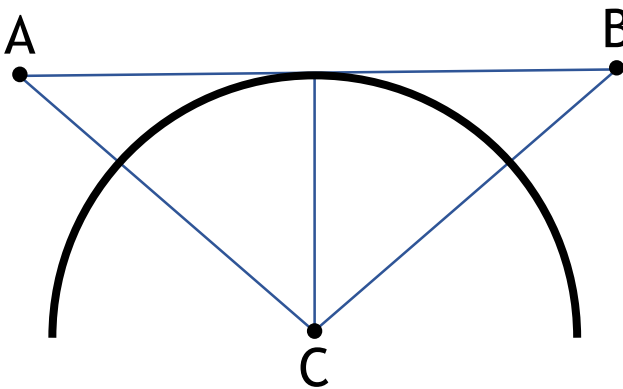
Сравним сумму углов $(a + b)$ и угол c .

Найдем его через теорему косинусов: $c = \arccos\left(\frac{AC^2 + BC^2 - AB^2}{2 * AC * BC}\right)$

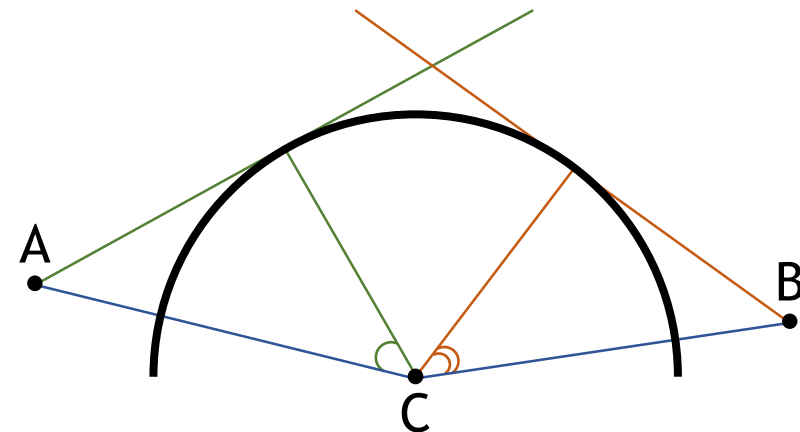
Схематично изобразим возможные варианты:



$$(a + b) > c$$



$$(a + b) = c$$



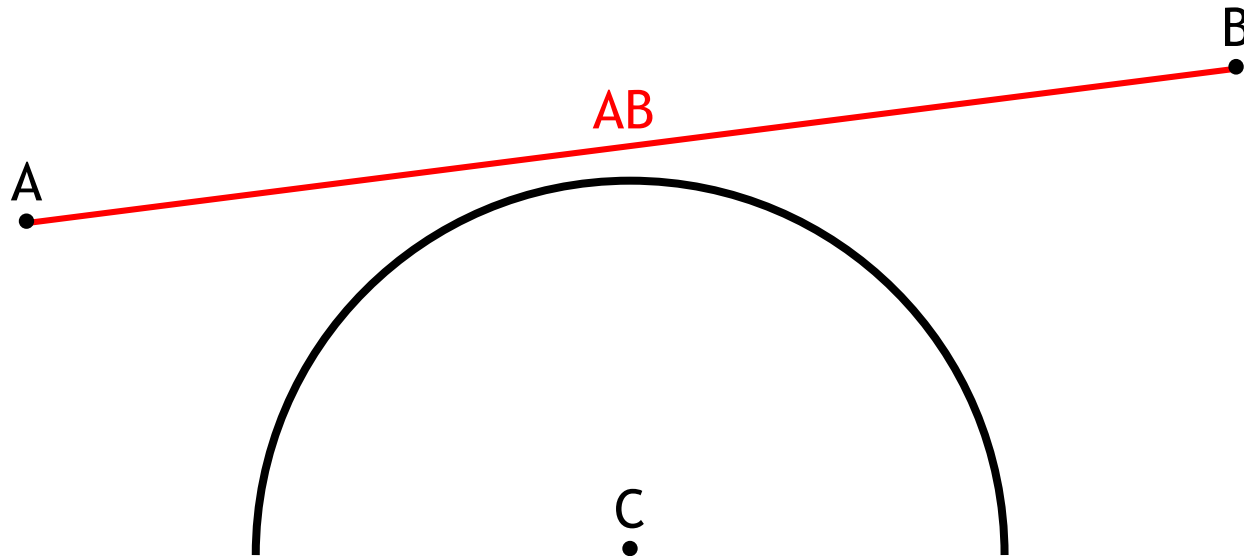
$$(a + b) < c$$

Нитка идет по прямой

Следовательно, нитка идет по прямой в двух случаях:

1. $(a + b) > c$
2. $(a + b) = c$

Длина такой нити будет равна расстоянию АВ:

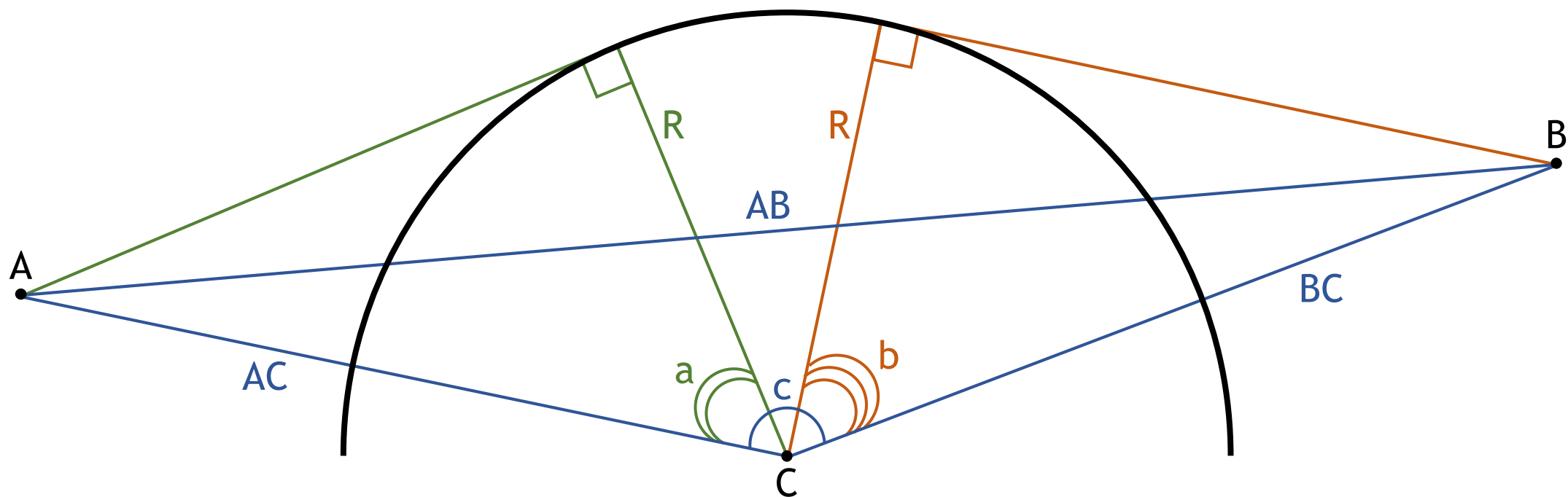


Нитка идет по шару

Найдем расстояния АВ, АС и ВС.

Затем, вычислим углы а, b и с.

Здесь $(a + b) < c$, следовательно, нитка идет по шару



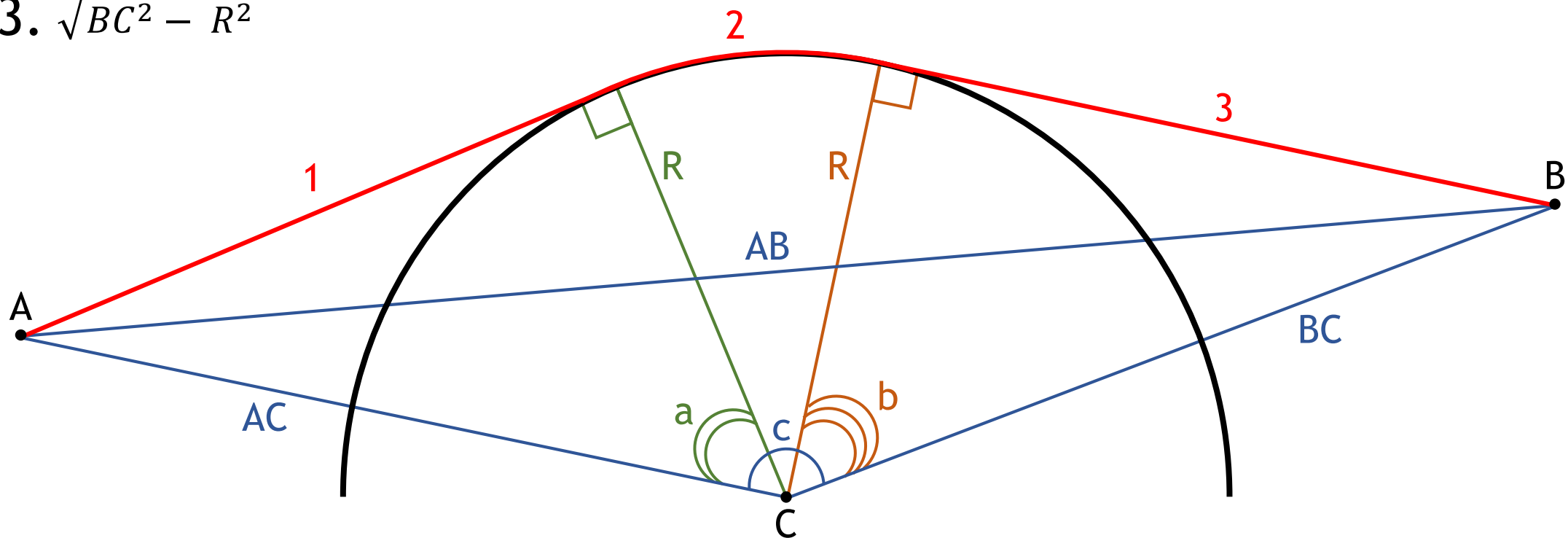
Нитка идет по шару

Длину нити мы найдем как сумму трёх отрезков:

1. $\sqrt{AC^2 - R^2}$

2. $(c - a - b) * R$

3. $\sqrt{BC^2 - R^2}$



Алгоритм на языке Rust

```
fn find_thread_length(a: Point, b: Point, c: Point, r: f64) -> f64 {  
    let line_ab = get_distance(&a, &b);  
    let line_ac = get_distance(&a, &c);  
    let line_bc = get_distance(&b, &c);  
  
    let angle_c = acos((sqr(line_ac) + sqr(line_bc) - sqr(line_ab)) / (2.0 * line_ac * line_bc));  
    let angle_a = acos(r / line_ac);  
    let angle_b = acos(r / line_bc);  
  
    if angle_a + angle_b >= angle_c {  
        return line_ab;  
    } else {  
        return (angle_c - angle_a - angle_b) * r  
            + sqrt(sqr(line_ac) - sqr(r))  
            + sqrt(sqr(line_bc) - sqr(r));  
    }  
}
```

Вопросы