Julia Grzegorzewska, Wiktoria Fimińska

Link do repozytorium: https://github.com/grzesiaaa/Algorytmy_lista2

RAPORT LISTA 2

Znalazłyśmy 5 sposobów na znalezienie trójki pitagorejskiej o podanym obwodzie trójkąta. Każda z nich w kolejności jest bardziej optymalna od poprzedniej.

Pierwszy z nich został zaprezentowany na zajęciach. Polega on na tym, że każdy bok jest poszukiwany w pętli spośród liczb mniejszych od obwodu trójkąta. Następnie sprawdzane są zależności $a^2 + b^2 = c^2$ oraz $a + b + c = l$, gdzie l to obwód trójkąta . Jeśli wartości spełniają je, program wyrzuca krotkę z True, długościami boków i liczbą operacji.

```
def pythagorean_triple_1(length):
    operations = 0
    for a in range(1, length):
        for b in range(1, length):
            for c in range(1, length):
                operations += 9
                if a ** 2 + b ** 2 == c ** 2 and a + b + c == length:
                    return True, a, b, c, operations
    return False, None, None, None, operations
```

W następnym korzystamy już tylko z dwóch pętli, a trzeci bok otrzymujemy z różnicy obwodu i sumy reszty boków. Zakładamy, że a<=b, dlatego b lecimy w pętli już od a zamiast od 1. Jeśli boki spełniają równanie $a^2 + b^2 = c^2$ to funkcja wyrzuca krotkę taką, jak w poprzednim programie.

```
def pythagorean_triple_2(length):
    operations = 0
    for a in range(1, length):
        for b in range(a, length):
            operations += 7
            c = length - a - b
            if a ** 2 + b ** 2 == c ** 2:
                return True, a, b, c, operations
    return False, None, None, None, operations
```

W trzecim ograniczamy poszukiwane wartości do połowy wartości obwodu i działamy analogicznie jak w poprzednim.

```
def pythagorean_triple_3(length):
    operations = 0
    for a in range(1, length//2):
        for b in range(a, length//2):
            operations += 7
            c = length - a - b
            if a ** 2 + b ** 2 == c ** 2:
                return True, a, b, c, operations
    return False, None, None, None, operations
```

Czwarty program rozwiązuje problem najszybciej. Pierwszy bok poszukiwany jest w pętli od 1 do jednej trzeciej części obwodu (a<b<c), a dwa kolejne są uzależnione od niego oraz obwodu (wyliczenia pokazane na poniższym zdjęciu). Tym razem przy sprawdzaniu zależności $a^2 + b^2 = c^2$ nie zapisujemy potęgi jako „a**2” lecz „a*a” i dzięki temu program działa szybciej.

```
def pythagorean_triple_4(length):
    operations = 0
    for a in range(1, length//3):
        operations += 14
        b = (2*a*length-length*length)//(2*(a-length))
        c = length - a - b
        if a * a + b * b == c * c:
            return True, a, b, c, operations
    return False, None, None, None, operations
```

W piątym przykładzie wzorowałyśmy się na sposobach wyliczania trójek z internetu. M zaczyna się od 2, ponieważ m>n>0. Metoda ta szuka „pierwszych” trójek pitagorejskich, więc aby móc znaleźć wszystkie wprowadzamy liczbę k („złożone” trójki tworzymy poprzez „wielokrotność” tych pierwszych).

```python
def pythagorean_triple_5(length):
    c = 0
    m = 2
    operations = 0
    while c < length//2:
        for n in range(1, m):
            operations += 13
            a = m * m - n * n
            b = 2 * m * n
            c = m * m + n * n
            primitive_length = a + b + c
            if length % primitive_length == 0:
                operations += 5
                k = length // primitive_length
                a *= k
                b *= k
                c *= k
                return True, a, b, c, operations
        m += 1
    return False, None, None, None, operations
```

W każdym przykładzie liczona jest też liczba operacji.

Dla każdej z funkcji sprawdziłyśmy czas jej wykonania dla length = 90 (dla większego length ciężko by było je porównać, ponieważ pierwsza funkcja jest bardzo wolna). Poniżej wyniki.

Nazwa funkcji	Liczba operacji	Czas wykonania
pythagorean_triple_1	601920	0.034354041528701786 s
pythagorean_triple_2	5012	0.0003429825305938721 s
pythagorean_triple_3	2492	0.00017083234786987304 s
pythagorean_triple_4	126	1.5979766845703125e-06 s
Pythagorean_triple_5	44	7.998466491699219e-07

Pomocne linki:

https://www.cuemath.com/geometry/pythagorean-triples/