

Lista 2

Maciej Karczewski

zad 2 i 3

Zad. 2 Trójka pitagorejska to trzy całkowite liczby dodatnie a, b i c spełniające równanie

$$a^2 + b^2 = c^2$$

Istnieje tylko jedna trójka taka, że

$$a + b + c = 1000$$

Znajdź ohe.

Zad. 3 Podaj liczbę działań potrzebnych do rozwiązania poprzedniego zadania.

• Początkowa wersja programu, która działa bardzo nie wydajnie dla

lepiej tego nie sprawdzać.

$$a + b + c = 1000$$

```
def pit(s):
    steps = 0
    for a in range(1,s):
        for b in range(1,s):
            for c in range(1,s):
                steps += 8
                if a**2 + b**2 == c**2 and a + b + c == s:
                    return True, a, b, c, steps
    return False, -1, -1, -1, steps
```

```
In [2]: print(pit(1000))
print(pit(1001))

(True, 200, 375, 425, 1591810000)
(False, -1, -1, -1, 8000000000)
```

• Wiemy, że gdy szukana suma to "s" to

$$c = s - a - b$$

```
def pit2(s):
    steps = 0
    for a in range(1,s):
        for b in range(1,s):
            c = s - a - b
            steps += 7
            if a**2 + b**2 == c**2:
                return True, a, b, c, steps
    return False, -1, -1, -1, steps
```

```
In [3]: print(pit2(1000))
print(pit2(1001))

(True, 200, 375, 425, 1394232)
(False, -1, -1, -1, 7000000)
```

• Wiemy że

$$a + b + c = s \Rightarrow b < s - a$$

```
def pit3(s):
    steps = 0
    for a in range(1,s):
        for b in range(1, s - a):
            c = s - a - b
            steps += 7
            if a**2 + b**2 == c**2:
                return True, a, b, c, steps
    return False, -1, -1, -1, steps
```

```
In [4]: print(pit3(1000))
print(pit3(1001))

(True, 200, 375, 425, 1254932)
(False, -1, -1, -1, 3496500)
```

• Wiemy z nierówności trójkąta, że

$$a < \frac{s}{2}$$

```
def pit4(s):
    steps = 0
    for a in range(1, s // 2):
        for b in range(1, s - a):
            c = s - a - b
            steps += 7
            if a**2 + b**2 == c**2:
                return True, a, b, c, steps
    return False, -1, -1, -1, steps
```

```
In [5]: print(pit4(1000))
print(pit4(1001))

(True, 200, 375, 425, 1254932)
(False, -1, -1, -1, 2619750)
```

• Wiemy, że

oraz

Włęc

$$c = s - a - b$$
$$a + b > c$$
$$a + b > s - a - b \Rightarrow 2a + 2b > s \Rightarrow b > \frac{s - 2a}{2}$$

```
def pit5(s):
    steps = 0
    for a in range(1, s // 2):
        for b in range((s - 2*a)//2, s - a):
            c = s - a - b
            steps += 7
            if a**2 + b**2 == c**2:
                return True, a, b, c, steps
    return False, -1, -1, -1, steps
```

```
In [6]: print(pit5(1000))
print(pit5(1001))

(True, 200, 375, 425, 697032)
(False, -1, -1, -1, 1749993)
```

• Dodałkowo

$$a < c \Rightarrow a < s - a - b \Rightarrow b < s - 2a$$

```
def pit6(s):
    steps = 0
    for a in range(1, s // 2):
        for b in range((s - 2*a)//2, s - 2*a):
            c = s - a - b
            steps += 7
            if a**2 + b**2 == c**2:
                return True, a, b, c, steps
    return False, -1, -1, -1, steps
```

```
In [7]: print(pit6(1000))
print(pit6(1001))

(True, 200, 375, 425, 557732)
(False, -1, -1, -1, 876743)
```

• Dodatkowo

z nierówności wyżej

mamy

```
def pit7(s):
    steps = 0
    for a in range(1,s // 3):
        for b in range((s- 2*a)//2 , s - 2*a):
            c = s - a -b
            steps += 7
            if a**2 + b**2 == c**2:
                return True, a, b, c, steps
    return False , -1, -1, -1, steps
```

In [8]:

```
print(pit7(1000))
print(pit7(1001))

(True, 200, 375, 425, 557732)
(False, -1, -1, -1, 777378)
```

$$b < c \Rightarrow b < s - a - b \Rightarrow a < s - 2b$$
$$b < s - 2a$$
$$a < s - 2s + 4a \Rightarrow a < \frac{s}{3}$$

Podsumowanie

Porównanie ilości kroków dla sumy długości boku równej 1000

In [9]:

```
print("Ilość kroków programu szukającego trójki pitagorejskiej takiej ,że a + b + c = 1000 ")
print("Pierwsza wersja programu: " + str(pit(1000)[4]))
print("Druga wersja programu: " + str(pit2(1000)[4]))
print("Trzecia wersja programu: " + str(pit3(1000)[4]))
print("Czwarta wersja programu: " + str(pit4(1000)[4]))
print("Piąta wersja programu: " + str(pit5(1000)[4]))
print("Szósta wersja programu: " + str(pit6(1000)[4]))
print("Siódma wersja programu: " + str(pit7(1000)[4]))
```

Ilość kroków programu szukającego trójki pitagorejskiej takiej ,że a + b + c = 1000
Pierwsza wersja programu: 1591810000
Druga wersja programu: 1394232
Trzecia wersja programu: 1254932
Czwarta wersja programu: 1254932
Piąta wersja programu: 697032
Szósta wersja programu: 557732
Siódma wersja programu: 557732

Porównaie ilości kroków dla niemożliwej sumy długości boków równej 1001

In [11]:

```
print("Ilość kroków programu szukającego trójki pitagorejskiej takiej ,że a + b + c = 1001 , która nie istnieje. ")
print("Pierwsza wersja programu: " + str(pit(1001)[4]))
print("Druga wersja programu: " + str(pit2(1001)[4]))
print("Trzecia wersja programu: " + str(pit3(1001)[4]))
print("Czwarta wersja programu: " + str(pit4(1001)[4]))
print("Piąta wersja programu: " + str(pit5(1001)[4]))
print("Szósta wersja programu: " + str(pit6(1001)[4]))
print("Siódma wersja programu: " + str(pit7(1001)[4]))
```

Ilość kroków programu szukającego trójki pitagorejskiej takiej ,że a + b + c = 1001 , która nie istnieje.
Pierwsza wersja programu: 8000000000
Druga wersja programu: 7000000
Trzecia wersja programu: 3496500
Czwarta wersja programu: 2619750
Piąta wersja programu: 1749993
Szósta wersja programu: 876743
Siódma wersja programu: 777378

Kod

```

In [1]: def pit(s):
        """Function check if exist and give sizes of right triangle which have sum of side equal s
        @param s: (int) sum of length of sides
        @return: (bool, int, int, int, int) (True ,a ,b ,c ,k) if tringle exist otherwise (False ,-1 ,-1 ,-1 ,k) where a, b and c are length of this tringle and k is nessesary steps"""
        steps = 0
        for a in range(1,s):
            for b in range(1,s):
                for c in range(1,s):
                    steps += 8
                    if a**2 + b**2 == c**2 and a + b + c == s:
                        return True, a, b, c, steps
        return False, -1, -1, -1, steps

def pit2(s):
        """Function check if exist and give sizes of right triangle which have sum of side equal s
        @param s: (int) sum of length of sides
        @return: (bool, int, int, int, int) (True ,a ,b ,c ,k) if tringle exist otherwise (False ,-1 ,-1 ,-1 ,k) where a, b and c are length of this tringle and k is nessesary steps"""
        steps = 0
        for a in range(1,s):
            for b in range(1,s):
                c = s - a - b
                steps += 7
                if a**2 + b**2 == c**2:
                    return True, a, b, c, steps
        return False, -1, -1, -1, steps

def pit3(s):
        """Function check if exist and give sizes of right triangle which have sum of side equal s
        @param s: (int) sum of length of sides
        @return: (bool, int, int, int, int) (True ,a ,b ,c ,k) if tringle exist otherwise (False ,-1 ,-1 ,-1 ,k) where a, b and c are length of this tringle and k is nessesary steps"""
        steps = 0
        for a in range(1,s):
            for b in range(1,s - a):
                c = s - a - b
                steps += 7
                if a**2 + b**2 == c**2:
                    return True, a, b, c, steps
        return False, -1, -1, -1, steps

def pit4(s):
        """Function check if exist and give sizes of right triangle which have sum of side equal s
        @param s: (int) sum of length of sides
        @return: (bool, int, int, int, int) (True ,a ,b ,c ,k) if tringle exist otherwise (False ,-1 ,-1 ,-1 ,k) where a, b and c are length of this tringle and k is nessesary steps"""
        steps = 0
        for a in range(1,s // 2): # a + b > c
            for b in range(1,s - a):
                c = s - a - b
                steps += 7
                if a**2 + b**2 == c**2:
                    return True, a, b, c, steps
        return False, -1, -1, -1, steps

def pit5(s):
        """Function check if exist and give sizes of right triangle which have sum of side equal s
        @param s: (int) sum of length of sides
        @return: (bool, int, int, int, int) (True ,a ,b ,c ,k) if tringle exist otherwise (False ,-1 ,-1 ,-1 ,k) where a, b and c are length of this tringle and k is nessesary steps"""
        steps = 0
        for a in range(1,s // 2):
            for b in range((s - 2*a)//2, s - a): # a + b > c => b > (s-2a)/2
                c = s - a - b
                steps += 7
                if a**2 + b**2 == c**2:
                    return True, a, b, c, steps
        return False, -1, -1, -1, steps

def pit6(s):
        """Function check if exist and give sizes of right triangle which have sum of side equal s
        @param s: (int) sum of length of sides
        @return: (bool, int, int, int, int) (True ,a ,b ,c ,k) if tringle exist otherwise (False ,-1 ,-1 ,-1 ,k) where a, b and c are length of this tringle and k is nessesary steps"""
        steps = 0
        for a in range(1,s // 2):
            for b in range((s - 2*a)//2, s - 2*a): # a < c
                c = s - a - b
                steps += 7
                if a**2 + b**2 == c**2:
                    return True, a, b, c, steps
        return False, -1, -1, -1, steps

def pit7(s):
        """Function check if exist and give sizes of right triangle which have sum of side equal s
        @param s: (int) sum of length of sides
        @return: (bool, int, int, int, int) (True ,a ,b ,c ,k) if tringle exist otherwise (False ,-1 ,-1 ,-1 ,k) where a, b and c are length of this tringle and k is nessesary steps"""
        steps = 0
        for a in range(1,s // 3): # b < c
            for b in range((s - 2*a)//2, s - 2*a):
                c = s - a - b
                steps += 7
                if a**2 + b**2 == c**2:
                    return True, a, b, c, steps
        return False, -1, -1, -1, steps

```

In []: