

Lista 2

Maciej Karczewski

zad 2 i 3

**Zad. 2** Trójką pitagorejska to trzy całkowite liczby dodatnie  $a, b$  i  $c$  spełniające równanie  $a^2 + b^2 = c^2$

Istnieje tylko jedna trójką 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

$$a + b + c = 1000$$

lepiej tego nie sprawdzać.

```
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 [21]: 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):
        steps += 1
        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 [22]: print(pit3(1000))
print(pit3(1001))
(True, 200, 375, 425, 1255132)
(False, -1, -1, -1, 3497500)
```

- Wiemy z nierówności trójkąta, że

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

```
def pit4(s):
    steps = 1
    for a in range(1,s // 2):
        steps += 1
        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 [23]: print(pit4(1000))
print(pit4(1001))
(True, 200, 375, 425, 1255133)
(False, -1, -1, -1, 2620250)
```

- Wiemy, że

$$c = s - a - b$$

oraz

$$a + b > c$$

Więc

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

```
def pit5(s):
    steps = 1
    for a in range(1,s // 2):
        steps += 4
        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 [24]: print(pit5(1000))
print(pit5(1001))
(True, 200, 375, 425, 698033)
(False, -1, -1, -1, 1752489)
```

- Dodałkowio

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

```
def pit6(s):
    steps = 1
    for a in range(1,s // 2):
        steps += 5
        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 [25]: print(pit6(1000))
print(pit6(1001))

(True, 200, 375, 425, 558933)
(False, -1, -1, -1, 879738)
```

• Dodatkowo

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

z nierówności wyżej

$$b < s - 2a$$

mamy

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

```
def pit7(s):
    steps = 1
    for a in range(1,s // 3):
        steps += 5
        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 [26]: print(pit7(1000))
print(pit7(1001))

(True, 200, 375, 425, 559133)
(False, -1, -1, -1, 779783)
```

Podsumowanie

Porównanie ilości kroków dla sumy długości boku równej 1000

```
In [27]: 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: 1255132  
Czwarta wersja programu: 1255133  
Piąta wersja programu: 698033  
Szósta wersja programu: 568933  
Siódma wersja programu: 559133

Porównanie ilości kroków dla niemożliwej sumy długości boków równej 1001

```
In [28]: 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: 3497500  
Czwarta wersja programu: 2620250  
Piąta wersja programu: 1752400  
Szósta wersja programu: 879738  
Siódma wersja programu: 779703

Kod

[https://github.com/maciejkarlookig\\_for\\_pitagoras\\_triplets.git](https://github.com/maciejkarlookig_for_pitagoras_triplets.git) ([https://github.com/maciejkarlookig\\_for\\_pitagoras\\_triplets.git](https://github.com/maciejkarlookig_for_pitagoras_triplets.git))

```

In [20]: 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):
        steps += 1
        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 = 1
    for a in range(1,s // 2): # a + b > c
        steps += 1
        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 = 1
    for a in range(1,s // 2):
        steps += 5
        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 = 1
    for a in range(1,s // 2):
        steps += 6
        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 = 1
    for a in range(1,s // 3): # b < c
        steps += 7
        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 [ ]: