

js → Python  
 == → ===  
 stara wersja, nowa wersja

```
zadanie.py > ...
1 def containsDuplicate(nums):
2     seen = set()
3
4     for num in nums:
5         if num in seen:
6             return True
7         seen.add(num)
8     return False
9
10 result = containsDuplicate([1, 2, 3, 1])
11 print(result) # Output: True
```

Variable

A  
B  
...  
Z

key = ?

[Left | right] 1  
 ↑ ↑  
 1 2  
 3-4

CPU

$$n \times n = n^2$$

↑  
time

RAM

Set

↑  
space (memory)

Storage

SSD / HDD  
1TB

1000 000

$$O(n) = O(N)$$

$O(n)$

$O(n)$

Set  
100.003 size

for (i=0; i<n; i++)

[1, 2, 3, 100000, 1, ..... 100000]

$$\log_2 8 = 3 \Rightarrow 2^3 = 8$$

$$O(\log n) = O(\log_2 n)$$

$$\log_2 1000000 = 19,93$$

$$2^{19,93} = 1000000$$

20 операций VS 1000 000

Set

• add  
• delete  
• has

→  $O(1)$

[1, 4, 6, 8] [9, 11, 13, 17]

↑ 2 3

↑ 1

$$\log 8 = 3$$

target = 6

$O(n)$  — это обозначение асимптотической сложности алгоритма в информатике ("О большое от n"). Оно означает, что время выполнения (или память) растёт линейно пропорционально размеру входных данных n. В коде функции contains\_duplicate цикл проходит по каждому элементу списка один раз, поэтому сложность —  $O(n)$ . Это эффективно для больших n, в отличие от  $O(n^2)$  или хуже