Lab 2

Luca Novello | Feb 3, 2025 | DSA456V1A

# Function 1

| def function1(number):  total=0  for i in range(0, number):  x = (i+1)  total += (x \* x)  return total |
| --- |

## Step 1: Establish Variables and Functions

* Let n represent the input number.
* Let T(n) represent the number of operations required to return the total.

## Step 2: Count your Operations

* **total = 0:** 1 operation
* **for i in range(0, number):** n iterations
* **x = (i + 1)** n operations
* **total += (x \* x)** n operations (multiplication + addition)
* **return total** 1 operation

**Total: 1 + n + n + n + 1 = 3n + 2**

## Step 3: Establish Mathematical Expression

T(n)=3n+2

## Step 4: Simplify the Equation

The highest-order term is 3n, and constants are ignored in Big-O notation.

## Step 5: State your Final Result

Therefore, T(n) is O(n).

# Function 2

| def function2(number):  return ((number)\*(number+1)\*(2\*number + 1))/6 |
| --- |

## Step 1: Establish Variables and Functions

* Let n represent the input number.
* Let T(n) represent the number of operations.

## Step 2: Count your Operations

* function performs a fixed number of multiplications, additions, and a division at constant rate.

Total operation count is **O(1)**

## Step 3: Establish Mathematical Expression

T(n)=O(1)

## Step 4: Simplify the Equation

T(n)=O(1)

## Step 5: State your Final Result

T(n)=O(1)

# Function 3

| def function3(list):  for i in range (0,len(list)-1):  for j in range(0,len(list)-1-i):  if(list[j]>list[j+1]):  tmp=list[j]  list[j]=list[j+1]  list[j+1]=tmp |
| --- |

## Step 1: Establish Variables and Functions

* Let n represent the length of lst.
* Let T(n) represent the number of operations needed for sorting.

## Step 2: Count your Operations

* Outer loop runs **(n-1) times**.
* Inner loop runs **(n-1), (n-2), (n-3), ..., 1** **times**.
* Swap operation happens for each comparison.

**Total: T(n)=O(n2)**

## Step 3: Establish Mathematical Expression

T(n)=O(n2)

## Step 4: Simplify the Equation

T(n)=O(n2)

## Step 5: State your Final Result

T(n)=O(n2)

# Function 4

| def function4(number):  total = 1  for i in range(1, number):  total \*= (i + 1)  return total |
| --- |

## Step 1: Establish Variables and Functions

* Let n represent the input number.
* Let T(n) represent the number of operations.

## Step 2: Count your Operations

* total = 1 → 1 operation.
* for i in range(1, number): → (n-1) iterations.
  + total \*= (i + 1) → (n-1) multiplications

**Total: 1 + (n-1) + (n-1) = 2n - 1**

## Step 3: Establish Mathematical Expression

T(n)=2n−1

## Step 4: Simplify the Equation

* The highest-order term is n.
* Constants are ignored.

## Step 5: State your Final Result

T(n)=O(n)