

CSC108H Winter 2024 Worksheet 32.5 (Optional) : Run Time Complexity of Functions

L refers to a list. Make a chart plotting the length of the list as the x axis and the number of times the assignment statement `sum = sum + i + j` is executed as the y axis. Plot for lengths 0, 1, 2, 3, 4, and 5.

```
sum = 0
for i in range(len(L)):
    for j in range(i):
        sum = sum + i + j
```

Does this code have constant, linear, or quadratic running time?

(a) constant (b) linear (c) quadratic

Now consider the following functions and choose the best description of the runtime for each of them:

```
1. def miles_per_gallon(rods: float, hogsheads: float) -> float:
    """Return the fuel efficiency in mpg for a distance travelled, rods, and
    a volume of gas consumed, hogsheads.
    """
    yards_travelled = rods * 5.5
    miles_travelled = yards_travelled // 1760
    gallons_consumed = hogsheads * 63
    return miles_travelled / gallons_consumed
```

Complexity: _____ in terms of rods + hogsheads

```
2. def weakly_fluffy(w: str) -> bool:
    """Return True if and only iff word is "weakly fluffy". A weakly fluffy word
    has at least one fluffy character (a character appearing in 'fluffy').
    """
    for character in w:
        if character in 'fluffy':
            return True
    return False
```

Complexity: _____ in terms of len(w)

```
3. def unfluffy(w: str) -> bool:
    """Return True if and only if no characters in the given word are fluffy.
    """
    return not weakly_fluffy(w)
```

Complexity: _____ in terms of len(w)

```
4. def approximate_mean(L: list[float]) -> float:
    """Return an approximation of the mean of a sorted list, as the halfway
    point between the smallest and largest values.
    """
    smallest = L[0]
    largest = L[-1]
    return (smallest + largest) / 2
```

Complexity: _____ in terms of len(L)

CSC108H Winter 2024 Worksheet 32.5 (Optional) : Run Time Complexity of Functions

```
5. def biggest_powerof2_divisor(n: int) -> int:
    """Return the largest power of 2 that divides n.
    """
    # The current largest divisor. 1 divides every integer.
    divisor = 1
    while n % 2 == 0:
        n = n // 2
        divisor = divisor * 2
    return divisor
```

Complexity: _____ in terms of n

```
6. def square_pairs(L: list[int]) -> int:
    """Return the number of "square pairs" among numbers in the list. A square pair
    consists of two numbers where one is the square of the other.
    """
    pairs = 0
    for num1 in L:
        for num2 in L:
            if num1**2 == num2:
                pairs = pairs + 1
            elif num2**2 == num1:
                pairs = pairs + 1

    return pairs
```

Complexity: _____ in terms of len(L)

```
7. def second_smallest(L: list) -> object:
    """Return the second-smallest item in L.
    Precondition: len(L) >= 2
    """
    smallest = min(L)
    L.remove(smallest)
    return min(L)
```

Complexity: _____ in terms of len(L)