Name- ศิรชนก ศิริวลพัวณ์ ID 1670700044

Section 227E

Homework 1: Analyzing Time Complexity (Big O)

Examine the code and count how many times the operations are executed. Then, detail step by step how to determine the Big O notation.

1.

```
def sum_of_squares(n):
    total = 0 1
    i = 1  1
    while i <= n:
        total += i * i
        i += 1
    return total 1
```

4n+1

Count (4n+1)+1+1+1

4n+4

Operation count:

= : 1 operations

= : 1 operations

while

<=, +=, *, += : 4 operations

return : 1 operations

Count 1+1+(4n+1)+1 = 4n+4

Step-by-Step to Find Big O

f(n) = 4n+4

f(n) = 4n+4n

f(n) = 8n

f(n) = O(n), C(8), n_0 = ?

n1 n_0

4n+4n = 8n

4n = 4n

n = 1

Ans f(n) = O(n), C(8), n_0 = 1

2.

```
def example1(n):
    i = 0 1
    while i < n: n+1
        j = 0  1
        while j < 1: 2 *n 5n
            j += 1 1
        i += 1 1
```

Count 1+(n+1)+5n

6n+2

Operation count:

1
n+1
1n
2n
1n
1n

6n+2

Step-by-Step to Find Big O

f(n) = 6n+2

f(n) = 6n+2n

f(n) = 8n

f(n) = O(n), C(8), n_0 = ?

n1 n_0

6n+2n = 8n

2n = 2n

n = 1

Ans f(n) = O(n), C(8), n_0 = 1

3.

```
def example1(n):
    i = 0
    while i < n:
        j = 0
        while j < n:
            print(i, j)
            j += 1
        i += 1
```

Count = $1+(n+1)+n+(2n^2+n)+n$

$= 2n^2+4n+2$

Operation count:

1
$n+1$
n
$n+1$
n
n

$2n^2+4n+2$

Step-by-Step to Find Big O

$f(n) = 2n^2+4n+2$

$f(n) = 2n^2+4n^2+2n^2$

$f(n) = 8n^2$

$f(n) = O(n^2), C=8, n_0 = ?$

At n_0: $2n^2+4n+2n^2 = 8n^2$

$2n^2 = 2n^2$

$n = 1$

Ans $f(n) = O(n^2), C=8, n_0 = 1$

4.

```
def nested_loop_linear(n):
    i = 0
    while i < 3:
        j = 0
        while j < n:
            print("i =", i, "j =", j)
            j += 1
        i += 1
```

Count = $1+4+3+15+12+3$

$= 38$

Operation count:

1
4
3
15
12
3

38

Step-by-Step to Find Big O

$f(n) = O(1), C=38, n_0 = 1$

5.

```
def nested_loop_linear():
    i = 1
    total = 0
    while i < 3:
        j = 1
        while j < 10:
            total += j * i
            j += 1
        i += 1
    return total
```

Count 4+74+3=84

Operation count:

1
1
3
2
10
18
9
8
1

Step-by-Step to Find Big O

$f(n) = O(1), C(84), n_0 = 1$

6.

```
def cubic_function(n):
    total = 0
    i = 0
    while i < n:
        j = 0
        while j < n:
            k = 0
            while k < n:
                total += 1
                k += 1
            j += 1
        i += 1
    return total
```

Count = $1+1+(n+1)+n+n+(n^2+n)+n^2+n^2+(n^3+n^2)+n^3+n^3+1$

$= 3n^3+4n^2+4n+4$

Operation count:

1
1
(n+1)
n
(n^2+n)
n^2
(n^3+n^2)
n^3
n^3
n^2
n
1

Step-by-Step to Find Big O

$f(n) = 3n^3+4n^2+4n+4$
$f(n) = 3n^3+4n^3+4n^3+4n^3$
$f(n) = 15n^3$
$f(n) = O(n^3), C(15), n_0 = ?$

$n_0 = ?$

$3n^3+4n^3+4n^3+4n^3 = 15n^3$
$12n^3 = 12n^3$
$n = 1$

Ans $O(n^3), C(15), n_0 = 1$

```
def extra_challenge(n):
    total = 0 1
    i = 1 1
    while i <= n: n+1
        j = 1 n
        while j <= i:
            total += 1
            j += 1
        i += 1 n
    return total 1
```

3n+1 x n
3n²+n

Count = 1+1+(n+1)+n+(3n²+n)+n+1
= 3n²+4n+4

Operation count:

1
1
n+1
n
n+1
n
n
n
n

3n²+4n+4

Step-by-Step to Find Big O

f(n) = 3n²+4n+4
f(n) = 3n²+4n²+4n²
f(n) = 11n²
f(n) = O(n²), C(8), n₀ = ?

n₀
3n²+4n²+4n² = 11n²
8n² = 8n²
n = 1
Total O(n²), C(8), n₀ = 1