ECS 32B - Introduction to Data Structures - Homework 2.5

Name: Zonglin Han

Student ID: 925223470

Assumptions for time analysis:

1. Counting variable not included;
2. One statement = one operation;

Problem 1

(1)

```
def fun1(n):
    x = 0
    for i in range(n/2):
        x = 2*5 - i
    return x
```

`x = 0` $\rightarrow 1$

loop $\rightarrow \lfloor \frac{n}{2} \rfloor$

`return` $\rightarrow 1$

$T(n) = \lfloor \frac{n}{2} \rfloor + 2$

Therefore

$T(n) \in O(n)$

Proof:

$\forall n \geq 1, T(n) = \lfloor \frac{n}{2} \rfloor + 2 \leq \frac{n}{2} + 2 \leq 3n$

Choose $c = 3, n_0 = 1$.

Hence $T(n) \in O(n)$.

(2)

```
def fun2(n):
    x = 0
    for i in range(1,n):
        for j in range(1,n):
            for k in range(1,n):
                x = x + 1
    return x
```

`x = 0` $\rightarrow 1$

Each loop runs from 1 to $n-1$, totally $n-1$ times

There's three nested loops, and the inner loop body executes one statement so totally $(n-1)^3$ times

`return` $\rightarrow 1$

$$T(n) = (n-1)^3 + 2 = n^3 - 3n^2 + 3n - 1 + 2 = n^3 - 3n^2 + 3n + 1$$

Therefore

$$T(n) \in O(n^3)$$

Proof:

$$\forall n \geq 1, T(n) = n^3 - 3n^2 + 3n + 1 \leq n^3 + 3n^3 = 8n^3$$

Choose $c = 8$, $n_0 = 1$.

Hence $T(n) \in O(n^3)$.

(3)

```python
def fun3(n):
    x = 0
    i = n
    while i >= 0:
        x = x - i
        i = i - 1
    return x
```

`x = 0` $\rightarrow 1$

`i = n` $\rightarrow 1$

Loop runs for $n+1$ times, and there are 2 statements in the loop body, so totally $2(n+1)$ times.

`return` $\rightarrow 1$

$$T(n) = 1 + 1 + 2(n+1) + 1 = 2n + 5$$

Therefore

$$T(n) \in O(n)$$

Proof:

$$\forall n \geq 1, T(n) = 2n + 5 \leq 7n$$

Choose $c = 7$, $n_0 = 1$.

Hence $T(n) \in O(n)$.

(4)

```
def fun4(n):
    x = n
    for i in range(n):
        x = x + 1
    for i in range(n * n):
        x = x + 2
    return x
```

`x = n` $\rightarrow 1$

First loop runs for n times, and there is 1 statement in the loop body, so totally n times.

Second loop runs for n^2 times, and there is 1 statement in the loop body, so totally n^2 times.

`return` $\rightarrow 1$

$T(n) = n^2 + n + 2$

Therefore

$T(n) \in O(n^2)$

Proof:

$\forall n \geq 1, T(n) = n^2 + n + 2 \leq 4n^2$

Choose $c = 4$, $n_0 = 1$.

Hence $T(n) \in O(n^2)$.

(5)

```
def fun5(lst):
    x = 0
    for y in lst:
        x = x + y
    return x
```

Suppose the input list have length n.

`x = 0` $\rightarrow 1$

Loop runs once per element $\rightarrow n$

`return` $\rightarrow 1$

$T(n) = n + 2$

Therefore

$T(n) \in O(n)$

Proof:

$\forall n \geq 1, T(n) = n + 2 \leq 3n$

Choose $c = 3, n_0 = 1$.

Hence $T(n) \in O(n)$.

Problem 2

(1)

$T(n) = 5$

Therefore

$T(n) \in \Theta(1)$

Proof:

$\forall n \geq 1, 5 \cdot 1 \leq 5 \leq 5 \cdot 1$

Choose $c_1 = 5, c_2 = 5, n_0 = 1$.

Then $c_1 \cdot 1 \leq T(n) \leq c_2 \cdot 1$.

Hence $T(n) = \Theta(1)$.

(2)

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

Therefore

$T(n) \in \Theta(n^2)$

Proof:

$\forall n \geq 1, n^2 \leq 2n^2 + 1 \leq 3n^2$

Choose $c_1 = 1, c_2 = 3, n_0 = 1$.

Then $c_1 n^2 \leq T(n) \leq c_2 n^2$.

Hence $T(n) \in \Theta(n^2)$.

(3)

$T(n) = 3n^4 + 2n^3 + 2n$

Therefore

$T(n) \in \Theta(n^4)$

Proof:

$\forall n \geq 1, 3n^4 \leq 3n^4 + 2n^3 + 2n \leq 3n^4 + 2n^4 + 2n^4 = 7n^4$

Choose $c_1 = 3, c_2 = 7, n_0 = 1$.

Then $c_1 n^4 \leq T(n) \leq c_2 n^4$.

Hence $T(n) \in \Theta(n^4)$.

(4)

$T(n) = \log(5 \times 2^n) = \log 5 + n \log 2$

Therefore

$T(n) \in \Theta(n)$

Proof:

$\forall n \geq 1, n \log 2 \leq \log 5 + n \log 2 \leq (\log 5 + \log 2)n$

Choose $c_1 = \log 2, c_2 = \log 5 + \log 2, n_0 = 1$.

Then $c_1 n \leq T(n) \leq c_2 n$.

Hence $T(n) \in \Theta(n)$.

(5)

$T(n) = 4n^2 \log n$

Therefore

$T(n) \in \Theta(n^2 \log n)$

Proof:

$\forall n \geq 1, 4n^2 \log n \leq 4n^2 \log n \leq 4n^2 \log n$

Choose $c_1 = 4, c_2 = 4, n_0 = 2$.

Then $c_1 n^2 \log n \leq T(n) \leq c_2 n^2 \log n$.

Hence $T(n) = \Theta(n^2 \log n)$.