Matt Langlois
7731812

CSI2110A - Assignment 1

1 a) $T(N) = c$ $\therefore$ $T(2N) = c$

$500 = c$ $\quad T(2N) = 500$

b) $T(N) = cN$ $\therefore$ $T(2N) = c2N$

$500 = cN$

$$= 2cN$$
$$= 2(500)$$
$$= 1000$$

c) $T(N) = cN^2$ $\therefore$ $T(2N) = c(2N)^2$

$500 = cN^2$

$$= c4N^2$$
$$= 4cN^2$$
$$= 4(500)$$
$$= 2000$$

d) $T(N) = 2^N$ $\therefore$ $T(2N) = 2^{2N}$

$500 = 2^N$

$$= (2^N)^2$$
$$= 500^2$$
$$= 250{,}000$$

2 a) $(n+1)^5$ is $O(n^5)$

$(n+1)^5 = n^5 + 5n^4 + 10n^3 + 10n^2 + 5n + 1$

$$n^5 + 5n^4 + 10n^3 + 10n^2 + 5n + 1 \leq n^5 + 5n^5 + 10n^5 + 10n^5 + 5n^5 + n^5 \quad \text{for all } n \geq 1$$
$$\leq 32n^5$$

$\therefore n_0 = 1 \quad c = 32$

b) 3^{n+2} is $O(3^n)$

$3^n \cdot 3^2 \leq (1+3^2) \cdot 3^n$

$3^n \cdot 9 \leq 10 \cdot 3^n$ for all $n \geq 0$

c) n^2 is $\Omega(n \cdot \log n)$

$n \geq \log_2(n)$

$n^2 \geq \log_2(n^2)$

$n \cdot n \geq n \log(n)$

$\therefore$ $c = 1$ $n_0 = 1$

d) n^3 is $O(n^2)$

Assume $n^3 \leq cN^2$

then $n \leq c$ for all n

$\therefore$ this false by Contradiction because c is not always larger than n

hence $n^3 \neq O(n^2)$

e) $\log(n^{10} + n^2)$ is $O(\log(n))$

$$\log(n^{10} + n^2) \leq \log(n^{10} + n^{10})$$
$$\leq \log(2n^{10})$$
$$\leq 1 + 10 \log n$$
$$\leq \log(n) + 10 \log(n)$$
$$\leq 11 \log(n)$$

$\log(n) \leq 1$

$\therefore C = 11$ and $n_0 = 2$

③ Calculating the number of swaps for 3:

$$2(3) + 2(3)(3-1) + 2(3)(3-1)(3-2)$$
$$= 2(3+6+6) \leftarrow 2 \text{ can be factored out}$$
$$= 2(15)$$
$$= 30$$

Calculating the number of swaps for n:

$$F(0) = 0$$
$$F(1) = 2$$
$$F(2) = 2(2) + 2(2)(2-1) = 8$$
$$F(3) = 2(3) + 2(3)(3-1)(3-2) = 30$$

2 swaps each iteration — Initial call — first call — n^{th} recursive call

$$F(n) = 2(n) + 2(n)(n-1) + \dots + 2(n!) = 2(\underbrace{n + n(n-1) + \dots + n!}_{n \text{ times}})$$

Big-O Notation:

$$2(\underbrace{n + n(n-1) + \dots + n!}_{n \text{ times}}) \leq 2(n! + n! + \dots + n!) \quad \forall\, n \geq 1$$
$$\leq 2(n(n!)) \quad n_0 \geq 1$$
$$c = 2$$

$\therefore$ the recursive function is $O(n(n!))$ for all $n \geq 1$ and $c=2$.

4 a)

$P_1 = \{a, b, c, d, e\}$

$P_2 \{\}$

$F = \{\}$

After first loop

$P_1 = \{\}$

$P_2 = \{e, d, c, b, a\}$

$F = \{\}$

After final loop

$P_1 = \{a, b, c, d, e\}$

$P_2 = \{\}$

$F = \{\}$

b)

$P_1 = \{a, b, c, d, e\}$

$P_2 = \{\}$

$F = \{\}$

After first loop

$P_1 = \{\}$

$P_2 = \{\}$

$F = \{e, d, c, b, a\}$

After final loop

$P_1 = \{e, d, c, b, a\}$

$P_2 = \{\}$

$F = \{\}$

c)

```java
public static void swap(Stack<String> stack, int i, int j) {
    Queue<String> q = new LinkedList<>();
    Stack<String> s = new Stack<>();
    int size = stack.size();
    for (int n = 0; n < size; n++) {
        if (n==i || n == j) {
            q.add(stack.pop());
        } else {
            s.push(stack.pop());
        }
    }
    for (int n = 0; n < size; n++) {
        if (n==i || n == j) {
            stack.push(q.poll());
        } else {
            stack.push(s.pop());
        }
    }
}
```