# Exercise Sheet 2

Mengxuan Wu

September 26, 2023

## Question 2

### Question 2.1

$$
\begin{aligned}
3n^2 + 5n - 2 = \Theta(n^2) &\qquad c_1 = 2 \quad c_2 = 4 \quad n_0 = 5 \\
42 = \Theta(1) &\qquad c_1 = 41 \quad c_2 = 43 \quad n_0 = 1 \\
4n^2 \cdot (1 + \log n) - 2n^2 = \Theta(n^2 \log n) &\qquad c_1 = 3 \quad c_2 = 5 \quad n_0 = 4
\end{aligned}
$$

### Question 2.2

| $f(n)$ | $g(n)$ | $O$ | $o$ | $\Omega$ | $\omega$ | $\Theta$ |
|---|---|---|---|---|---|---|
| $\log n$ | $\sqrt{n}$ | yes | yes | no | no | no |
| $n$ | $\sqrt{n}$ | no | no | yes | yes | no |
| $n$ | $n \log n$ | yes | yes | no | no | no |
| $n^2$ | $n^2 + (\log n)^3$ | yes | no | yes | no | yes |
| $2^n$ | $n^3$ | no | no | yes | yes | no |
| $2^{n/2}$ | $2^n$ | yes | yes | no | no | no |
| $\log_2 n$ | $\log_{10} n$ | yes | no | yes | no | yes |

### Question 2.3

For algorithm $A$, line 1 is executed once and line 4, 5 and 6 are executed $n^2 - 2n$ times each. The number of foo operation is $3n^2 - 6n + 1 = \Theta(n^2)$, as $2n^2 \leq 3n^2 - 6n + 1 \leq 4n^2$ for all $n \geq 6$.

For algorithm $B$, line 1 is executed once, line 3 is executed $n$ times and line 5 and 6 are executed $n/2$ times each. The number of foo operation is $2n + 1 = \Theta(n)$, as $n \leq 2n + 1 \leq 3n$ for all $n \geq 1$.

For algorithm $C$, line 1 and 6 are executed once each, line 4 is executed $n(n+1)/2$ times and line 5 is executed $n$ times. The number of foo operation is $\frac{1}{2}n^2 + \frac{3}{2}n + 2 = \Theta(n^2)$, as $\frac{1}{2}n^2 \leq \frac{1}{2}n^2 + \frac{3}{2}n + 2 \leq n^2$ for all $n \geq 4$.

## Question 2.4

1. True.

*Proof.* For all $f(n) \in O(\sqrt{n})$, there exists some $c, n_0 > 0$ that satisfy $0 \leq f(n) \leq c\sqrt{n}$ for all $n \geq n_0$. Since $0 \leq c\sqrt{n} \leq cn$ holds for all positive integer $n$, we can infer that $0 \leq f(n) \leq c\sqrt{n} \leq cn$ for all $n \geq n_0$, with the same $c$ and $n_0$. Hence, we have $f(n) \in O(n)$. $\square$

2. False.

*Proof.* Since $n = o(n^2)$ holds, we can infer that $n + n = 2n = \omega(n)$. However, $\lim_{n\to\infty} \frac{2n}{n} = 2 \neq \infty$. There is a contradiction. $\square$

3. True.

*Proof.* Fisrtly, we will proof $3n \log n + O(n) = O(n \log n)$. Since there exists some $c_0, n_0 > 0$ that satisfy $0 \leq 3n \log n + O(n) \leq 3n \log n + c_0 n$ for all $n \geq n_0$, we can deduce that $0 \leq 3n \log n + O(n) \leq 3n \log n + c_0 n < 3n \log n + c_0 n \log n = (3 + c_0) n \log n$. Hence, we have $3n \log n + O(n) = O(n \log n)$, with $c_1 = c_0 + 3$ and $n_1 = n_0$.

Secondly, we will proof $3n \log n + O(n) = \Omega(n \log n)$. Since $O(n)$ has a non-negative value, we can infer that $0 \leq 3n \log n \leq 3n \log n + O(n)$. Hence, we have $3n \log n + O(n) = \Omega(n \log n)$, with $c_2 = 3$ and $n_2 = n_0$.

Finally, with two conclusions above, we can proof $3n \log n + O(n) = \Theta(n \log n)$. $\square$

4. The statement "The running time of Algorithm A is at least $O(n^2)$" is meaningless. Because notation $O$ means "at most" already, it contradicts "at least" before it.

## Question 2.5

| MATRIX-MULTIPLY$(A, B)$ | Runtime(in one iteration) |
|---|---|
| **for** $i = 1$ to $n$ **do** | $n + 1 = \Theta(n)$ |
|     **for** $j = 1$ to $n$ **do** | $n + 1 = \Theta(n)$ |
|         $C[i, j] = 0$ | $1 = \Theta(1)$ |
|         **for** $k = 1$ to $n$ **do** | $n + 1 = \Theta(n)$ |
|             $C[i, j] = C[i, j] + A[i, k] \cdot B[k, j]$ | $1 = \Theta(1)$ |
| **return** $C$ | $1 = \Theta(1)$ |

The total runtime of MATRIX-MULTIPLY is

$$\Theta(n) \cdot \Theta(n) \cdot (\Theta(1) + \Theta(n) \cdot \Theta(1)) + \Theta(1) = \Theta(n^3)$$