COMPSCI 2C03 – Week 3 Exercises

- These exercises are for independent study and to provide structure for your tutorials.
- A tutorial might cover one or two exercises at the most. It's a good idea to try the exercises yourself before tutorial, then you can bring questions to focus the session on what you most need help with.
- It's also a good idea to implement your solutions in Python or Java and then test thoroughly. This will help you find bugs and deepen your understanding of the material.

Lecture 1: Basic Sorting Algorithms

1. Compute the **worst case running time** function $T(n)$ for the algorithm below. Make sure to count all basic operations both inside and outside of the loop. The parameter **n** is the length of the array to search.

```
search(a: array, n: int, key):
      index ← 0
      found ← false
      while index < n and not found:
            if a[index] = key:
                  found = true
            index += 1
      return found
```

2. Compute the running time function $T(n)$ for the algorithm below. Make sure to count all basic operations both inside and outside of the loops. The **//** operator is **floor division**. It rounds down and returns the result as an integer.

```
sum ← 0
k ← n//2
while k > 0:
      for i←0 to k-1:
            sum ← sum + 1
      k ← k - 1
```

3. Write an insertion sort algorithm that works with a doubly linked list. When moving items within the list, you can assume it's ok just to move the contents of a node, you don't have to change the links to move the node itself.

Lecture 2: Algorithm Analysis

4. Using the definition of $O(f(n))$ and $\Omega(f(n))$, prove the following statements:
   a. $(70n^3 - 300n + 2)/2 \in O(n^3)$
   b. $12n^5 + n^4 - 700 \in O(n^6)$

c. $20n^2 - 33n - 22 \in \Omega(n^2)$
d. $11n^2 - 43 \in \Omega(n)$

5. Using the definition of $O(f(n))$ and $\Omega(f(n))$, prove the following statements:
   a. $(n^3 - 9n - 9)/2 \notin O(n^2)$
   b. $n^2 + n - 555 \notin O(n)$
   c. $11n^2 - 43 \notin \Omega(n^3)$

Lecture 3: Working with Big O

6. Write an algorithm that, given two sorted arrays of n numeric values, prints all elements that appear in both arrays, in sorted order. The running time of the program should be $O(n)$.

7. Use limits to prove the following:
   i. $12n^2 + 5$ is $\Theta(n^2)$
   ii. $3n^3 - 2n^2 + 5$ is $\Omega(n^2)$ but not $O(n^2)$
   iii. $5n^6 + 3n^3$ is $O(n^8)$ but not $\Omega(n^8)$

8. Use the six Useful Big O facts from the lecture to show that:

   $(n^2 + 5)(n - 4) + 15n \in O(1.5^n)$

9. Use the definition of Big O to prove Rule 1 from the “Big O Facts” slide:

   If $d(n) \in O(f(n))$ then $k \cdot d(n) \in O(f(n))$ for $k > 0$.

10. Use the definition of Big O (not limits or other properties) to prove the following:
    i. $10 \log n \in O(\ln n)$
    ii. $n^2 + 2 \log n \in O(n^2)$
    iii. $n \in (n \log n)$