CSC 225 SUMMER 2016
ALGORITHMS AND DATA STRUCTURES I
ASSIGNMENT 1
UNIVERSITY OF VICTORIA

1. Order the following list of functions by their big-Oh notation. Group together (for example, by underlining) those functions that are big-Theta of one another. (No justification needed).
   Note: $\log n = \log_2 n$ unless otherwise stated.

$6n \log n$	2^{100}	$\log \log n$	$\log^2 n$	$2^{\log n}$
2^{2^n}	$\sqrt{n}$	$n^{0.01}$	$1/n$	$4n^{3/2}$
$3n^{0.5}$	$5n$	$2n \log^2 n$	2^n	$n \log_4 n$
4^n	n^3	$n^2 \log n$	$4^{\log n}$	$\sqrt{\log n}$

   Hint: When in doubt about two functions $f(n)$ and $g(n)$, consider $\log f(n)$ and $\log g(n)$ or $2^{f(n)}$ and $2^{g(n)}$.

2. Justify the fact that if $d(n)$ is $O(f(n))$ and $e(n)$ is $O(g(n))$, then the product $d(n)e(n)$ is $O(f(n)g(n))$.

3. Show that $\log_b f(n)$ is $\Theta(\log_2 f(n))$ if $b > 1$ is a constant.
   Hint: One of the properties of logarithms from the slide I gave you is very useful here.

4. Consider the recurrence equation,

$$T(n) = \begin{cases} 1 & \text{if } n = 0 \\ T(n-1) + 2^n & \text{otherwise.} \end{cases}$$

   Show, by induction, that $T(n) = 2^{n+1} - 1$.

5. Consider the Algorithm `arrayFind`, given below, which searches an array A for an element x.

   Algorithm `arrayFind`(x, A):
   Input: An element x and an n-element array, A.
   Output: The index i such that $x = A[i]$ or -1 if no element of A is equal to x.

```
i ← 0
while i < n do
    if x = A[i] then
        return i
    else
        i ← i + 1
return −1
```

   (a) Counting assignments, comparisons, and returns only, calculate the worst-case, $T(n)$, and best-case, $T_b(n)$, running times of `arrayFind`.

   (b) Prove by induction (loop invariants) that `arrayFind` is correct.