CSC 225 SPRING 2021
ALGORITHMS AND DATA STRUCTURES I
MIDTERM EXAMINATION
UNIVERSITY OF VICTORIA

1. Student ID: __________________________

2. Name: ______________________________

3. DATE: 1 MARCH 2021
   DURATION: 50 MINUTES
   INSTRUCTOR: RICH LITTLE

4. THIS QUESTION PAPER HAS **FIVE** PAGES INCLUDING THE COVER PAGE.

5. THIS QUESTION PAPER HAS **FOUR** QUESTIONS.

6. ALL ANSWERS CAN BE WRITTEN ON THIS EXAMINATION PAPER, ON YOUR OWN PAPER OR TYPESET, PROVIDED YOU SUBMIT ONE PDF DOCUMENT CONTAINING THEM.

7. THIS IS AN OPEN BOOK EXAM. YOU MAY REFERENCE ANY COURSE MATERIALS; SLIDES, TEXTBOOKS, CONNEX.

8. NO OTHER ACCESS TO THE INTERNET IS ALLOWED, WITH THE POSSIBLE EXCEPTION OF AN ONLINE PDF APPLICATION.

9. NO COMMUNICATION/COLLABORATION ALLOWED WITH ANY OTHER PERSON.

10. YOUR SOLUTIONS MUST BE SUBMITTED IN THE ASSIGNMENTS TOOL IN CONNEX BY 5:30 PM VICTORIA TIME. EXAMS SUBMITTED IN ANY OTHER WAY WILL NOT BE ACCEPTED UNDER ANY CIRCUMSTANCES. LEAVE YOURSELF PLENTY OF TIME TO DO THE SUBMISSION PROCESS.

Q1 (8)	
Q2 (8)	
Q3 (8)	
Q4 (8)	
TOTAL (32) =	

1. (a). [3 marks] How many distinct positive integers n can we form using only the seven digits $2, 3, 4, 4, 5, 6, 7$, such that $n > 4{,}200{,}000$?

(b). [5 marks] I have $n = 3$ boxes of chocolate bars in my office, one with 24 Kit Kats, one with 24 Aero bars, and one with 24 Wonder bars. If I allow you to eat any r bars, such that there are 36 ways to select them, what is r?

2. (a). [4 Marks] Indicate for each pair of expressions (A,B) in the table below, whether A is O or Ω of B. Your answer should be in the form of a "yes" or "no" in each box.

A	B	O	Ω
$(\log n)^2$	n		
$2n^2 + 4\log n$	$4n^2 + n$		
4^n	2^n		
n^5/n	n^4		
100	log 100000		
$n^{1.5}$	$(1.5)^n$		

(b). [4 Marks] Show that if $p(n) = c_k n^k + c_{k-1} n^{k-1} + \cdots + c_2 n^2 + c_1 n + c_0$ is a polynomial in n, where $c_i \geq 0$ are real constants with $c_k > 0$ and $k \geq 1$ is a natural number, then $\log p(n)$ is $O(\log n)$.

3. (a) [3 Marks] Use induction to show that for all $n \geq 1$:

$$2 + 2^2 + 2^3 + \cdots + 2^n = 2^{n+1} - 2.$$

(b) [5 Marks] Solve the following recurrence equation to get a closed-formula for $T(n)$. You can assume the n is a power of two.

$$T(n) = \begin{cases} 1, & \text{if } n = 1 \\ T\left(\frac{n}{2}\right) + n, & \text{if } n \geq 2 \end{cases}$$

4. (a) [6 marks] I have written an amazing new sorting algorithm called `stupidSwapSort()` which sorts an array of n elements that have a total order. Below is the pseudocode for my sorting algorithm. Note that my algorithm does the two operations I have highlighted with a red rectangle every iteration no matter what the input. Count the runtime of my algorithm, $T(n)$, by counting the number of times those two commands occur.

```
Algorithm stupidSwapSort(A,n):
    Input: Array A of size n
    Output: Array A sorted

    for k ← 0 to n−2 do
      for j ← n−1 to k+1 do
        if A[j] > A[k] then
          swap(A[j], A[k])
        end
      end
    end
    for k ← 0 to n/2−1 do
      swap(A[k], A[n-1-k])
    end
end
```

(b) [2 marks] Based on my algorithm above you can see that the big-oh runtime in both the worst-case and best-case is the same. But, there is a best-case scenario in which my algorithm never performs the first swap command, `swap(A[j], A[k])`. Describe the input distribution in such a scenario, you may use an example.