NATIONAL UNIVERSITY OF SINGAPORE
SCHOOL OF COMPUTING

Midterm Assessment for CS3230

October 2, 2021 Time Allowed: 90 minutes

INSTRUCTIONS:

- This paper consists of **THREE** questions. Answer **ALL** questions. Please read instructions to each question carefully.
- 30 marks for each question. 10 marks for writing your matriculation number on the first page.
- This is an **OPEN BOOK/NOTES** examination.
- Do **NOT** use the internet for help while taking the examination. Any evidence otherwise will result in **severe penalties**.
- Do **NOT** seek the help of others during the examination. Any evidence otherwise will result in **severe penalties**.
- Please hand-write your solutions. Scan and submit as a **SINGLE PDF file** as instructed.

QUESTIONS:

1. (30 Marks)

   a) (10 Marks) Write true or false for each statement below. You **do not** need to give proofs. Here, lg denotes $\log_2$.

   (i)
$$\log_{100} n = \Theta\left(\sum_{i=1}^{n} \frac{1}{i}\right)$$

   (ii)
$$3^n = O(2^n)$$

   (iii)
$$\prod_{i=1}^{n} 2^i = o\left(n^{n^2}\right)$$

   (iv)
$$\lg n = \omega(1)$$

   b) (10 Marks) Use the substitution method to prove that $T(n) = O(\lg n)$ for $T(n)$ satisfying the following recurrence:
$$T(n) = 2T(\sqrt{n}) + 4.$$
   As usual, ignore the fact that $\sqrt{n}$ may not be an integer. Assume the base case that $T(n) = O(1)$ for $n \leq 5$.

   c) (10 Marks) Let $B : \{0,1\}^4 \to \{0,1\}$ be defined by $B(x_1, x_2, x_3, x_4) = (x_1 \texttt{ and } x_2) \texttt{ or } (x_3 \texttt{ and } x_4)$. Prove using the adversarial method that any deterministic algorithm for computing B must query all four input bits.

2. (30 Marks) Consider YetAnotherSort that is yet another sorting algorithm based on the divide-and-conquer strategy. The algorithm acts on an input array A of n distinct elements.

```
YetAnotherSort(A)
n = A.length
if n == 2 & A[1] > A[2]
      Swap A[1] and A[2]
elseif n > 2
      ℓ = ⌈2n/3⌉
      YetAnotherSort(A[1 .. ℓ])
      YetAnotherSort(A[n − ℓ + 1 .. n])
      YetAnotherSort(A[1 .. ℓ])
```

a) (20 Marks) Prove that the algorithm correctly sorts. Use induction to argue.

b) (10 Marks) State a recurrence (including the base cases) for the number of comparisons made by YetAnotherSort. Ignore the floors and ceilings. Solve the recurrence, and prove that your solution is correct.

3. (30 Marks)

   Given patterns $P_1, \ldots, P_k$ of equal length m, and text T of length n, design an algorithm that checks whether any of the patterns $P_1, \ldots, P_k$ occurs as a substring of T. All the strings are in binary.

   Your algorithm can be randomized with error probability $< 1\%$ and should minimize the expected running time. You should analyze both the correctness probability as well as the running time.

   Note: For this problem, assume that arithmetic on numbers representable in $O(\lg(nkm))$ bits can be performed in constant time.