**German University in Cairo**
**Media Engineering and Technology**
**Dr. Nourhan Ehab**

November 16, 2022

# CSEN 703: Analysis and Design of Algorithms
# Winter 2022
Midterm Exam

**Bar Code**

**Instructions: Read carefully before proceeding.**

1) Duration of the exam: 3 hours (180 minutes).
2) (Non-programmable) Calculators are allowed.
3) No books or other aids are permitted for this test.
4) This exam booklet contains 10 pages, excluding this one. Two extra sheets of scratch paper and a formula sheet are attached and have to be kept attached. **Note that if one or more pages are missing, you will lose their points. Thus, you must check that your exam booklet is complete**.
5) Write your solutions in the space provided. If you need more space, write the answer on the three extra sheets and make an arrow indicating that. **Scratch sheets will not be graded unless an arrow on the problem page indicates that the solution extends to the scratch sheets**.
6) When you are told that time is up, stop working on the test.

**Good Luck!**

---

Don't write anything below ;-)

| Exercise | 1 | 2 | 3 | 4 | 5 | $\sum$ |
|---|---|---|---|---|---|---|
| Possible Marks | 12 | 16 | 12 | 12 | 18 | 70 |
| Final Marks | | | | | | |

**Exercise 1** True/False Questions (2+2+2+2+2+2=12 Marks)

Decide whether each of the following statements is True or False. Justify your answer.

a) If $lim_{n\to\infty}(\frac{f(n)}{g(n)}) = c$, where c>0, then $f(n) = O(g(n))$ and $g(n) = O(f(n))$.

b) $(n+1)! = \Theta(n!)$

c) If $f(n) = O(g(n))$ and $g(n) = O(f(n))$ then $f(n) = g(n)$.

d) The best case running time of Merge Sort is different from its worst case running time.

e) A sorted array is the best case input for the Quick Sort algorithm.

f) The worst case running time of Quick Sort is asymptotically the same as its average case running time.

**Exercise 2** Bucket Sort (3+5+4+4=16 Marks)

Consider the following sorting algorithm referred to as *Bucket Sort*. In this exercise, assume that all arrays are 0-indexed and that the input array $A$ is composed of only positive numbers.

```
1  BucketSort(A, k)
2  buckets = new array of k empty arrays
3  max = maximum value in array A
4  for i = 0 to length(A) − 1 do
5  |  Insert A[i] in buckets[⌊ A[i]/max ∗ (k − 1)⌋]
6  end
7  for i = 0 to k − 1 do
8  |  Sort buckets[i]
9  end
10 return the concatenation of buckets[1], . . . , buckets[k]
```

a) Trace the operation of bucket sort given $A = [7, 2, 6, 1, 4, 3]$ and $k = 5$.

b) Prove that **BucketSort** is correct.

c) What is the best case scenario of **BucketSort**? What is the best case time complexity? Show all of your workout.

d) What is the worst case scenario of **BucketSort**? What is the worst case time complexity? Show all of your workout.

**Exercise 3** Recurrence Trees (8+4=12 Marks)

a) Obtain a **tight bound** on the running time of the following recurrence by using the recursion tree method. Draw the recursion tree and show all of your workout.

$$T(n) = 2T(\frac{n}{2}) + n\ log_2(n)$$

.

b) Can the above recurrence be solved using the master theorem? Justify your answer.

**Exercise 4** Master Theorem (3+3+3+3=12 Marks)

Can the following recurrences be solved using the master theorem? If yes, solve them. If not, explain why. Show all of your workout.

a) $T(n) = 16T(\frac{n}{4}) + n^3$

b) $T(n) = 5T(\frac{n}{2}) + n^2$

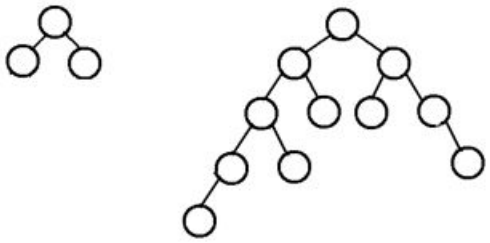
c) $T(n) = T(\frac{n}{4}) + 1$

d) $T(n) = 2T(\frac{n}{2}) + T(\frac{n}{4}) + n$

**Exercise 5** Divide and Conquer Algorithm Design (8+10=18 Marks)

a) Design a divide and conquer algorithm for computing the maximum number of levels in a binary tree.

For example: for the left tree below, the maximum number of levels is 2 while for the right tree, the maximum number of levels is 5. The algorithm must return 0 for empty trees and 1 for a single-node trees, respectively. You can use express your algorithm in English or using Pseudo Code. **Write the recurrence expressing the running time of your proposed algorithm.**



b) Given a 2D map with countries represented as points by their x and y coordinates, describe a divide and conquer algorithm to find the closest two countries. Your algorithm must be in $O(n\ log^2(n))$. You can express your algorithm in English or using Pseudo Code. **Write the recurrence expressing the running time of your proposed algorithm.**

## Useful Formulas

**Summations:**

- Constant series: $\sum_{i=j}^{k} a = a(k-j+1)$
- Arithmetic series: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$
- Finite Geometric series: $\sum_{i=0}^{n} r^i = \frac{1-r^{n+1}}{1-r}$

**Logarithms:**

- $ln(n) = log_e(n)$
- $log^k(n) = (log(n))^k$
- $log_c(ab) = log_c(a) + log_c(b)$
- $log_c(\frac{a}{b}) = log_c(a) - log_c(b)$
- $log_b(a^n) = nlog_b(a)$
- $a^{log_b(c)} = c^{log_b(a)}$
- Logarithmic change of base: $log_b(a) = \frac{log_c(a)}{log_c(b)}$
- $\frac{d}{dn} ln(n) = \frac{1}{n}$.
- $\frac{d}{dn} log_b(n) = \frac{1}{n\ ln(b)}$.

**Scratch Paper**

**Scratch Paper**