COMP2009 2022-23 ADE Coursework TWO (6.25%)
Wed. 29-MAR-2023
Time: 30 minutes.
Do not turn over the page until instructed.
Answer ALL FOUR questions for a total of 25 marks.
Calculators are not permitted.
Write your answers on these sheets within the spaces provided. Please write clearly. Write your name & ID in the box below CLEARLY AND IN UPPER CASE LETTERS

FAMILY NAME:	
FIRST NAME(S):	
Student ID number:	
Signature:	

(Also, write your name on each sheet; in case the sheets become separated.)

Information that might, or might not, be helpful:

Geometric series: $1 + 2 + 2^2 + 2^3 + \ldots + 2^p = 2^{p+1} - 1$

Powers of 2:

n	0	1	2	3	4	5	6
$\mathbf{2^n}$	1	2	4	8	16	32	64

Reminders of properties of logs:

$a^0 = 1$

log_2 (2^a) = a

log_b (a) = log_2 (a) / log_2 (b)

log_b (a) = 1 / log_a (b)

Master theorem: Given T(n) = a T(n/b) + f(n) and T(1)=1.

Case 1: If f(n) is O(n^c) for some c, with c < log_b(a)
then T(n) is Θ($n^{\log\_b(a)}$)

Case 2: If f(n) is Θ(n^c (log n) ^{k}) for some k ≥ 0, and with c = log_b (a)
then T(n) is Θ(n^c (log n)$^{k+1}$)

Case 3: If f(n) is Ω(n^c) for some c, with c > log_b(a)
then T(n) is Θ(f(n))
(strictly, we need f to satisfy a “regularity condition”, which you can ignore here)

For completion by markers:
Total mark (out of 25):

Question 1. "Vectors and Amortised complexity" [4 marks]

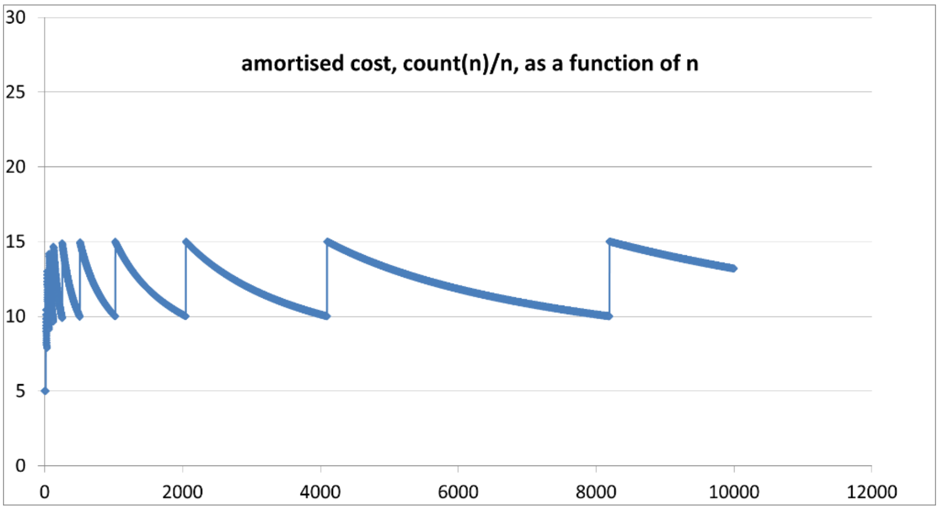
An empirical study of the amortised complexity of insertions into the Vector data structure is considered. The study consists of starting from a Vector data structure with just a small array and then inserting n extra elements one at a time – using a 'push' operation. An estimate of the total number of primitive operations is maintained:

> count(n) = estimate of the number of primitive operations (i.e. an estimate of the runtime) needed to push n elements starting from a small fixed size (a measure of the T(n) used in lectures)

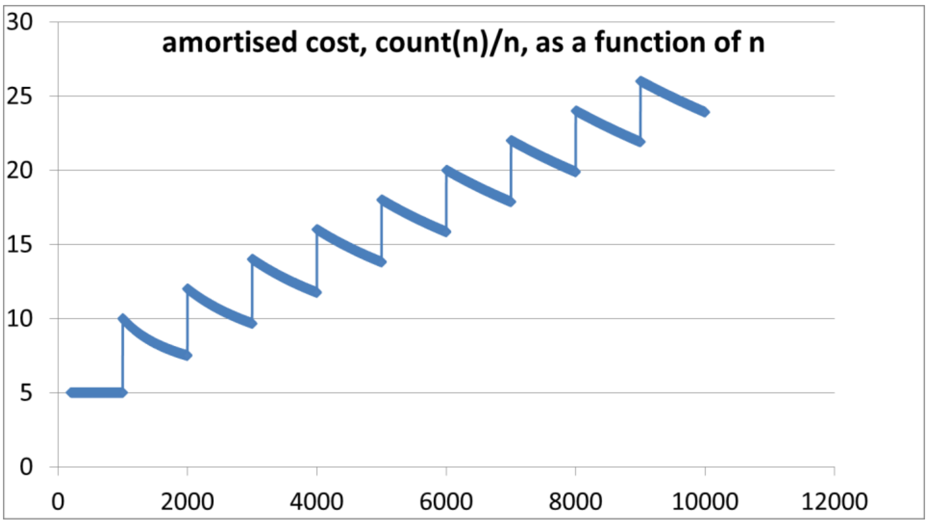
The amortised cost is then the function: count(n) / n
Two "resizing strategies" are studied for resizing the array when it is full:
- "**Incremental**" – increase the size by some constant number
- "**Doubling**" - double the size of the array each time

Two graphs below are obtained from plotting count(n)/n (y-axis) as a function of n (x-axis)

Graph A:



Graph B:



For each graph, clearly indicate which of the two resizing strategies it corresponds to (circle the correct answer and cross out the wrong one). Also, give a **brief** justification of your selection. You do NOT need to fully explain the graphs to get full marks!

Graph A	Circle one: "Incremental" or "Doubling" ? Brief Explanation:
Graph B	Circle one: "Incremental" or "Doubling" ? Brief Explanation:

Turn Over

Question 2. Recurrence – Master Theorem (MT) [7 marks]

Using the Master Theorem, identify the MT case, and solve for the Big-Theta behaviour of T(n) for the following three recurrence relations. In all three problems, you can assume T(1)=1. If using "Case 3" then you can assume that the regularity condition is satisfied. There is no need (or point) to justify your answers.

Q2.a **$T(n) = 8\ T(n/2) + n^2$**

Give the value of log_b (a):

Circle the case: Case 1 Case 2 Case 3

Circle the correct Theta behaviour of T(n):

A. $\Theta(n^2)$

B. $\Theta(n^2 \log(n))$

C. $\Theta(n^3)$

D. $\Theta(n^3 \log(n))$

E. $\Theta(n^4)$

F. $\Theta(n^4 \log(n))$

Q2.b $T(n) = 8\,T(n/2) + n^3$

Circle the case: Case 1 Case 2 Case 3

Circle the correct Theta behaviour of T(n):

A. $\Theta(n^2)$

B. $\Theta(n^2 \log(n))$

C. $\Theta(n^3)$

D. $\Theta(n^3 \log(n))$

E. $\Theta(n^4)$

F. $\Theta(n^4 \log(n))$

Q2.c $T(n) = 8\,T(n/2) + n^4$

Circle the case: Case 1 Case 2 Case 3

Circle the correct Theta behaviour of T(n):

A. $\Theta(n^2)$

B. $\Theta(n^2 \log(n))$

C. $\Theta(n^3)$

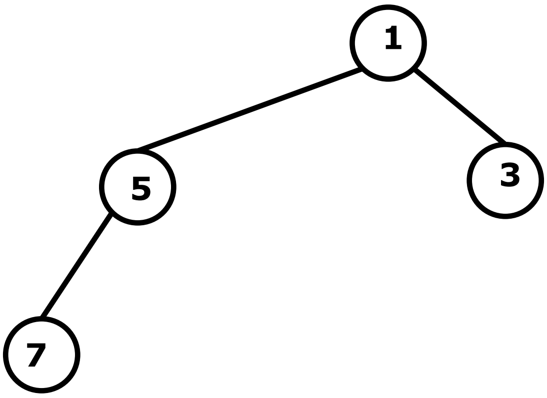
D. $\Theta(n^3 \log(n))$

E. $\Theta(n^4)$

F. $\Theta(n^4 \log(n))$

Question 3. Heaps [7 marks]

Consider the following heap:



Q3.a Complete the following array-based representation of the heap:

Array Index	0	1	2	3	4	5	6
Key	-	1					

Q3.b You are then to **insert the number 4** into the heap. Briefly explain the process, and give the heap (as a tree) that results after the insertion:

Q3.c You are then to **perform removeMin() on the tree that results from Q3.b**. Briefly explain the process, and give the heap (as a tree) that results after the removeMin.

Question 4. Recurrence relations [7 marks]

Consider the following recurrence relation:

$T(n) = 4\ T(\ n\ /\ 4\ ) + 1$ $\qquad\qquad$ $T(1) = 1$

The exact solution is claimed to be $T(\ 4^{k}\ ) = (\ 4^{k+1} - 1\ )\ /\ 3$
Use induction to prove that this solution is correct, and show your working.

Base Case:

Step Case:

END