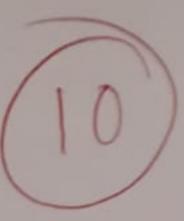
# Question 3: 11 points (4+3+4)



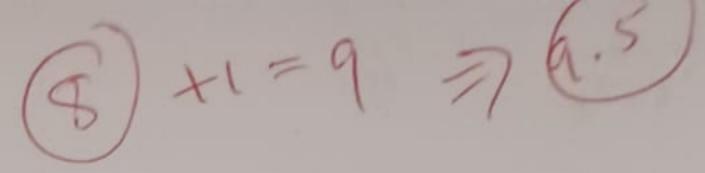
decided to sort the which of your Facebook friends are early adopters. So, you have decided to sort them using Facebook account ids which are 64-bit numbers. Which sorting algorithm will to Radix sort? E. ... Received account ids which are 64-bit numbers. Will be most appropriate – Insertion sort, Merge sort, Quicksort, Counting sort or Radix sort? Explain why.

Ans. Radize sort would be most appropriate. Because our datas account ids, is consisting of 64-bit numbers it would be good to sorte each digit starting from right most digit first. The 64-bit number consists 4 section as: 63-48/47-32/3F 16/15-We place each digit in every second and using Radiosort we sort it starting from right most to left.

### b. Use RadixSort, using LSB to MSB (IBM method) for the following array:

A = [455, 61, 63, 45, 67, 135, 74, 49, 15, 5] Ans. 063 35 455 455

## Question 1: 11 points (3+4+4)



a. Show the running time of the following code using big O notation (show your work): for (int 
$$j = 4$$
;  $j < n$ ;  $j=j+2$ ) {

$$val = 0;$$

$$for (int i = 0; i < j; ++i)$$

$$val = val + i * j;$$

$$for (int k = 0; k < n; ++k)$$

$$val++;$$

Ans. Hence it a nested for loop, will multiply
$$= (n/2+1)(4n-4)(3n-3)$$

$$= (2n^2-2n+4n-4)(3n-3)$$

$$= (2n^2+2n-4)(3n-3)$$

$$= (2n^2+2n-4)(3n-3)$$

$$= 6n^2+6n^2-6n-12n+12 = 0(6n^3-18n+12) = 0(n^3)$$

$$= 6n^3-6n^2+6n^2-6n-12n+12 = 0(6n^3-18n+12) = 0(6n^3-18n+$$

- b. Determine whether f is O(g), Theta (g) or Omega (g). Mention all that applies. Show your work.
- i) Show that  $2^{2n} != O(2^n)$  (!= means not equal to).

i) Show that 
$$2^{2n} != O(2^n)$$
 (!= means not equal to).

Ans.  $2^{2n} = 1$  im  $2^n = \infty$ , hence  $2^n != O(2^n)$  ?

Ans.  $2^n = 0$ 

(ii) 
$$f = n \wedge k$$
 and  $g = c \wedge n$ ;  $c > 1$  and  $k >= 1$ .

Ans. 
$$f = n^k$$
,  $g = e^n$ 

$$f(n) \times c g(n)$$

$$f($$

### Maharishi International University (MIU)

#### **MIDTERM**

Course Title and Code: CS 435 - Design and Analysis of Algorithms

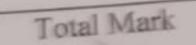
Instructor: Dr. Emdad Khan Date: Friday 07/29/2022 Duration: 10am - 12:30 pm

Student Name:

Amanuel Hadgu Tarete

Student ID:

614752



46

- 1. This is a closed book exam. Do not use any notes or books!
- Show your work. Partial credit will be given. Grading will be based on correctness, clarity and neatness.
- 3. We suggest you to read the whole exam before beginning to work on any problem.
- 4. There are 4 questions worth a total of 46 points, on 8 pages (including this one)
- You can use all back pages for scratch paper (may use for answers if you have used up the designated space for answer).
- 6. You can use a basic calculator (No smart phone unless network is disabled).

Question 2 (contd): 12 points (4+4+4)

c. Use Induction to prove that

$$\sum_{r=1}^{n} r(r+1) = \frac{1}{3}n(n+1)(n+2)$$

Ans.

First lets prove base case

=) 
$$\frac{1}{3}(1+1)(1+2) - \frac{1}{3}$$

ii) Assume this is true for k and let's prove for kell

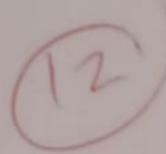
 $\frac{k}{k-1}(r+1) = \frac{1}{3}k(k+1)(k+2)$  // Assume

$$\frac{1}{2!} \frac{1}{r(r+1)} = \frac{\sum_{r=1}^{k} -(r+1) + (k+1)(k+1)+1}{\sum_{r=1}^{k} -(r+1) + (k+1)(k+2) + (k+1)(k+2)}$$

$$= \frac{1}{3} k(k+1)(k+2) + (k+1)(k+2)$$

$$=\frac{1}{3}(k+1)(k+2)(k+3)$$

## Question 2: 12 points (4+4+4)



For each of the following recurrences, derive an expression for the running time using iterative,

a. Consider the following recurrence algorithm [Use Master Theorem - See LAST Page]

Procedure (Array A, int n) {

(2 points) Write a recurrence equation for T(n)

ii. (2 points) Solve recurrence equation using Master's method i.e. give an expression for the runtime T(n).

ii. (2 points) Solve recurrence equation using Master's method i.e. give an expression for the runtime 
$$T(n)$$
.

Ans.  $a = 1$ 
 $b = 2$ 
 $c = 5$ 
 $c = 5$ 

Use Iterative method to solve the following recirsion:

$$T(n) = 3T(n-1) + 1, \quad T(1) = 0$$
Ans. 
$$T(n) = 3T(n-1) + 1$$

$$= 3[3T(n-2) + 1] + 1$$

$$= 3^{2}T(n-2) + 3 + 1$$

$$= 3^{2}[3T(n-3) + 1] + 3 + 1$$

$$= 3^{3}T(n-3) + 3^{2} + 3 + 1$$

$$T(k) = 3^{k}T(n-k) + 3^{k-1} + \dots + 3^{2} + 3 + 1$$
  
If  $k=n-1$ ,  $T(n-(n-1)) = T(1) = 0$   
 $= 3^{k}T(n-k) + \sum_{i=0}^{k-1} 3^{i}$  // from geometric sories  $\sum_{i=0}^{k-1} 3^{i} = 0$   
 $= 0 + 3^{k-1}$  //  $n=k+1$ 

#### Maharishi International University (MIU)

#### **MIDTERM**

Course Title and Code: CS 435 - Design and Analysis of Algorithms

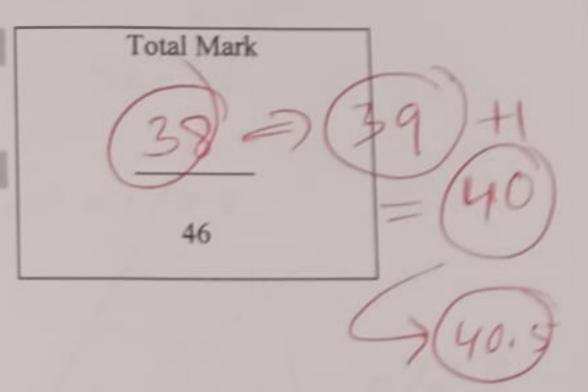
Instructor: Dr. Emdad Khan Date: Friday 07/29/2022 Duration: 10am - 12:30 pm

Student Name:

Amanuel Hadgu Tareke

Student ID:

614752



- 1. This is a closed book exam. Do not use any notes or books!
- Show your work. Partial credit will be given. Grading will be based on correctness, clarity and neatness.
- 3. We suggest you to read the whole exam before beginning to work on any problem.
- 4. There are 4 questions worth a total of 46 points, on 8 pages (including this one)
- You can use all back pages for scratch paper (may use for answers if you have used up the designated space for answer).
- 6. You can use a basic calculator (No smart phone unless network is disabled).

### Question 4: 12 points (3+5+4)



# a. Is Mathematics Consistent? Explain your answer with an example.

Ans. Mathematics is not consistent.

Hence mathematics can not proof all problems as both true and false. eg: 2° = 1

2' = 2

2' = 4

b.

### (i) Why do we need to use "Expected Value" in computing running time in Bucket sort?

Ans. We need to use "Expected Value" in computing running time in Bucket Sort, because the number of Buckets depends on the distribution of our data. Take array with values [1, 999,3." our Bucket number is n' < 5 < n = 10 < 5 103. 100,1217,6,502, 103,409,70 In the above array sorting we will need three 303,553 buckets of sige 10.

## (ii) Why Quicksort, in general performs better than Mergesort. Explain with an example using solutions for running time T(n) for both methods.

Ans. Ovicksort generally performs betten than Mergesort for the following reason:

(a) Outcksort is in-place, but Mergesort is not in place. This mergesort's use of extra space adds overhead to the operation which in turn slows the sorting in the worst case of Quicksort does NOT loccur often. This overcomes the weak side of Quicksort. Hence, the worst case our time of Quicksort (O(n²)) is not of much effect.

(11) Mergesort (T(n) = n logn), Quicksort (T(n) = n logn // querage most expected

c. Give a Big O estimate for

$$(2^n + n^2)(n^3 + 3^n)$$

Clearly show all steps.

Ans. 
$$(2^{n} + n^{2})(n^{3} + 3^{n})$$
  
=  $2^{n} \cdot n^{3} + 2^{n} \cdot 3^{n} + n^{2} \cdot n^{3} + n^{2} \cdot 3^{n}$   
=  $2^{n} \cdot n^{3} + 6^{n} + n^{5} + n^{2} \cdot 3^{n}$   
=  $0(2^{n} \cdot n^{3} + 6^{n} + n^{5} + n^{2} \cdot 3^{n})$   
=  $max(0(2^{n} \cdot n^{3}), 0(6^{n}), 0(n^{5}), 0(n^{2} \cdot 3^{n}))$   
=  $0(6^{n})$ 

algorithms cannot do better than  $O(n^2)$ .

Ans All inversion bound sorting algorithms can not do better than O(n')

Proof: The key operation in these algorithms is comparison. In

comparing we will always need nasted loop whose running

time is guadratic. So whenever we use these algorithms we

will have O(n²) running time.

Not a weed to we have a sorting to we have a sorting to the complete.

Not a proof.

#### Masters Formulae

For recurrences that arise from Divide-And-Conquer algorithms (like Binary Search), there is a general formula that can be used.

Theorem. Suppose T(n) satisfies

$$T(n) = \begin{cases} d & \text{if } n = 1\\ aT(\lceil \frac{n}{b} \rceil) + cn^k & \text{otherwise} \end{cases}$$

where k is a nonnegative integer and a, b, c, d are constants with  $a > 0, b > 1, c > 0, d \ge 0$ . Then

$$T(n) = \begin{cases} \Theta(n^k) & \text{if } a < b^k \\ \Theta(n^k \log n) & \text{if } a = b^k \\ \Theta(n^{\log_b a}) & \text{if } a > b^k \end{cases}$$

Question 3 (contd): 21 points (4+3+4)

E. Use the QuickSelect algorithm to manually compute the 4-th smallest element of the array [2, 5, what happens in each self-call, indicating the new input array and the current value of k.

Ans. A = [2,5,23,0,8,4,33,60,30] K = 4, E = [30] L2,5,23,0,8,4] E = [30] E = [30] E = [30]

121+1E1> K, the take L

[2,0] [5,23,3] [L]+|E| < k, Her take G, K' = K-|L|-|E| = 4-2-1

[23]
[23]
[1]+|E|< K, Hen take L

Ly Es] 6

Lorce, ILI < K & ILI + IEI

return 5