	COLLEGE OF COMPUTING AND INFORMATION SCIENCES
	FINAL Assessment Fall 2020 Semester

Class Id	105166/105167/ 105168/105272	**Course Title**	Design and Analysis of Algorithms
Program	BSCS	**Campus / Shift**	North Campus / Morning
Date	13th Dec 2020	**Total Points**	40
Duration	03 hours	**Faculty Name**	Dr. Salman Khan & Farooq Zaidi
Student Id	62357	**Student Name**	QasimHassan

INSTRUCTIONS:

- Students are expected to complete the exam within **03 hours**.
- To accommodate electricity or internet failures, students will be given **01 extra hour** for uploading and downloading.
- The exam will be from **1:00 PM to 4:00 PM**. Submission will be due at **5:00 PM**.
- Answers submitted after 5:00 pm will be penalized or even rejected.
- **Similar copies or material taken from the internet will be marked zero and plagiarism policy will be applied in accordance with the institutional regulations.**
- Students are advised to upload their answer scripts **as PDF file only** on **LMS** as soon as they finish the exam. A copy of the answer script should also be submitted to **Google classroom**.
- The answer script should be named in the following format: **ID_FirstName_CourseID.pdf**
- E.g. 12345_FirstName_105123.pdf, where 12345 is std id, and 105123 is replaced with your course id.
- Answers should be solved on this given question paper. No change should be made to the cover page of this exam paper.
- A recorded viva session may be conducted with students to ascertain the quality of answer scripts where deemed necessary.
- **The answer should preferably be typed**. Hand written answers (not recommended) should bear the name and ID number of the student on the page, and clear image of the page should be pasted in this exam sheet just below the concerned question.
- Any non-compliance with the given instructions may result in deduction of marks as penalty.

UNDERTAKING:

By attempting this exam, I declare that all answers in this examination have been given solely by me while using books, notes, internet and other sources of information, and I have not taken any help from any person or student in solving any part of this examination.

Question	Q1	Q2	Q3	Q4	Q5	Q6	Total
Max. marks	5	10 (5 +3+2)	8	7	5	5	40
Marks obtained							

Q1: WARM UP   [5 MARKS]

The following statements are FALSE. Provide the reason why they are FALSE (You answer should not exceed 10 words; longer answers will be penalized)

a. Simplicity, running time, space requirements, and algorithm structuring are all part of analysis.

It depends on the algorithms requirement

b. We have to look at all input sizes to make the order of growth of running time relevant.

we have to look at input sizes large enough only

c. Low order terms play a major role in defining the growth rate of a function.

no the growth function is determined by highest order term

d. Insertion sort and Merge sort both use divide-and-conquer approach.

No insertion sort uses Incremental **Approach**

e. Adjacency matrix takes O(V+E) storage.

adjacency matrix takes $O(|V|^2)$ memory space

Q2: ASYMPTOTIC ANALYSIS   [8 MARKS]

An algorithm A_1 has a runtime $T_1(n)$ defined (in seconds) by the following function:-

$$T_1(n) = 3n^2 + 70$$

For $n = 8$, the runtime of algorithm A_1 is **262** seconds.

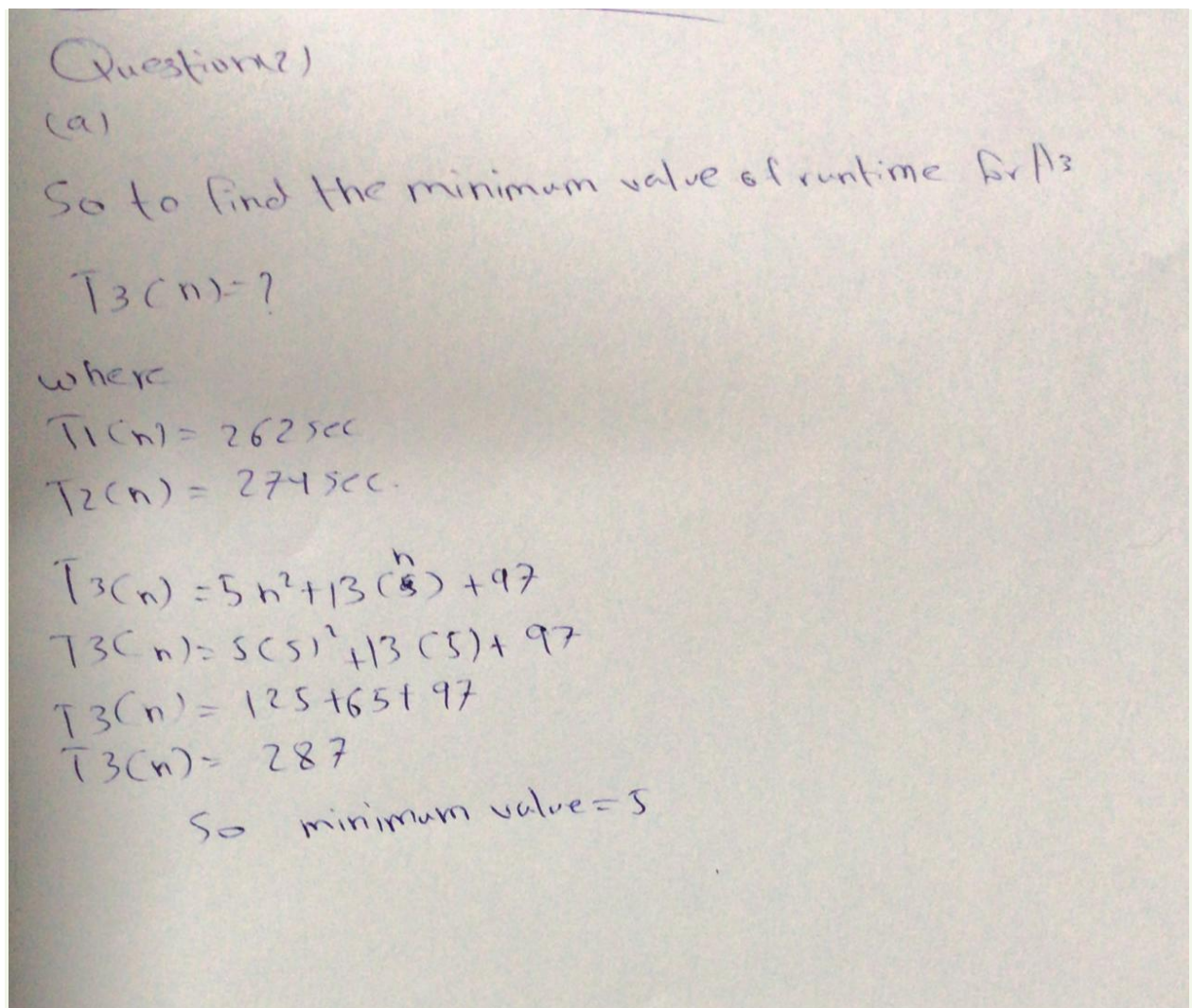Runtime of another algorithm A_2 is represented by the following equation:-

$$T_2(n) = 2n^3 + 14n + 230$$

For $n = 2$, the runtime of algorithm A_2 is **274** seconds.

Algorithm A_3 with runtime represented by the following equation.

$$T_3(n) = 5n^2 + 13n + 97$$

a. What is the minimum value of *n* for which the runtime of **A_3** exceeds the runtimes of A_1 and A_2 mentioned above?

Question2)

(a)

So to find the minimum value of runtime for A_3

$T_3(n) = ?$

where

$T_1(n) = 262 sec$

$T_2(n) = 274 sec.$

$T_3(n) = 5n^2 + 13(5) + 97$

$T_3(n) = 5(5)^2 + 13(5) + 97$

$T_3(n) = 125 + 65 + 97$

$T_3(n) = 287$

So   minimum value = 5

Consider the pseudo-code below:

```
01  F(n)
02  int sum = 0
03  for (int i = 0; i < n; i=i*2)
04       sum++
05  return n
```

b. Is the time efficiency of the code greater than, equal to, or less than O(n)?
Explain your answer.

Number of operations are less then n since the value of i is doubled in each
iterations so it takes O(log(n)) iterations for each value of i to reach n

c. What is the expected runtime of the code (in Big-O notation?). **Explain your answer**.

Run time of function is O(log(n))

Q3: GENETIC ALGORITHMS   [8 MARKS]

In a genetic algorithm, suppose your population contains 10 chromosomes. The size of each contains is 10. The fitness of a chromosome is calculated by the number of 1's it has. For example, if your chromosome is 1101101100, then its fitness F = 6.

a. For the ten chromosomes given in the table below, calculate their fitness

Chromosome	Fitness
1001001101	5
0110110011	6
1110111010	7
0101010101	5
1011001100	5
0000110010	3
1011001010	5
1101111011	8
1101110110	7
1100110011	6

b. Identify two fittest solutions and make them parents. Write down these parents

Parent 1 : **1101111011**
Parent 2 : **1110111010**

c. Using single-point crossover at the midpoint, perform crossover between the selected parents. Show the resulting offspring(s) and calculate its/their fitness

$P_1 = 11010111011$, $P_2 = 11101111010$

$P_1 = 11011 | 11011$
$P_2 = 11101 | 11010$

$C_1 = 11011\ 11010$, So fitness will be $= 7$

$C_2 = 11101\ 11011$, fitness will be $= 88$.

d. Now again assuming parents in part (b), perform a single point crossover, but select the cut point after the 4th gene. Show the resulting offspring(s) and calculate its/their fitness

Name :- Qasim Hassan

Std id :- G2357

Solution :-

$P_1 = 1101 | 111011$

$P_2 = 1110 | 111010$

$C1 = 110111010$

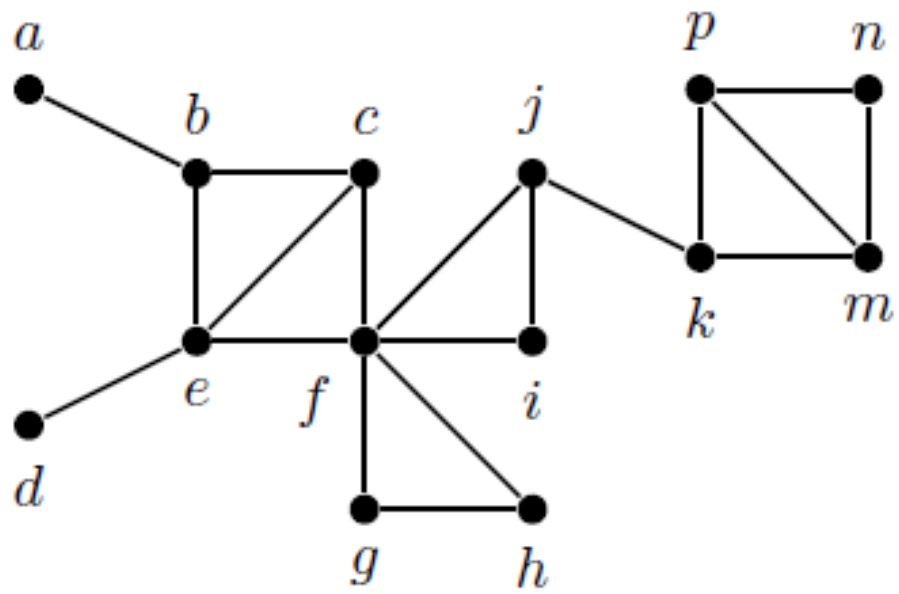$C2 = 111011011$

So fitness for $C1 = 7$

Fitness for $C2 = 8$

e. Using the results of part (c) and (d), choose the best offspring and perform mutation on gene 3 and gene 6. Write the resulting chromosome.

Type your answer here

Q4: GRAPH THEORY   [7 MARKS]

Consider the following graph.

a. How many different trees can be constructed considering f as the root node?

F → C → b → a (4 vertex, 3 edges)

F → e → b → a (4 vertex, 3 edges)

F → C → e → b → a (5 vertex, 4 edges)

F → C → e → d 8(4 vertex, 3 edges))

F → e → d (3 vertex, 2 edges)

F → C → b → e → a (5 vertex, 4 edges)

b. What will be the maximum depth of any tree found in part (a) above?

Mafimum depth

F→c→b→e→a

or

F→c→e→b→a

c. Construct the adjacency matrix for the graph given in the graph.

ADJACENCY MATRIX

	a	b	c	d	e	f	g	h	i	j	k	m	n	p
a	0	1	0	0	0	0	0	0	0	0	0	0	0	0
b	1	0	1	0	1	0	0	0	0	0	0	0	0	0
c	0	1	0	0	1	1	0	0	0	0	0	0	0	0
d	0	0	0	0	1	0	0	0	0	0	0	0	0	0
e	0	1	1	1	0	1	0	0	0	0	0	0	0	0
f	0	0	1	0	1	0	1	1	1	1	0	0	0	0
g	0	0	0	0	0	1	0	1	0	0	0	0	0	0
h	0	0	0	0	0	1	1	0	0	0	0	0	0	0
i	0	0	0	0	0	1	0	0	0	1	0	0	0	0
j	0	0	0	0	0	1	0	0	1	0	1	0	0	0
k	0	0	0	0	0	0	0	0	0	1	0	1	0	1
m	0	0	0	0	0	0	0	0	0	0	1	0	1	1
n	0	0	0	0	0	0	0	0	0	0	0	1	0	1
p	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Q5: IDENTIFYING THE TYPE OF PROBLEM   [5 MARKS]

The following table gives different problem instances. Identify whether the problem is a decision problem, function problem, or search problem?

Problem	Type
Finding the maximum value in given data	Searching
Finding longest path between two points in graph	Searching
Finding shortest path between two points	Searching
Finding square root of 81	Function
Finding factors of a prime number	Searching
Finding the variance for given data	Function
Finding whether a given integer is even number	Decision
Multiplication of two matrices	Function
Finding if 11 is greater than 9	Descision
Sorting an array with heap sort	Searching

Q6: DIVIDE & CONQUER AND DYNAMIC SOLUTION   [5 MARKS]

Assume you have a function f which is defined as follows:-

$$f(0) = 1$$
$$f(n) = f(floor(n/2)) + f(n-1) + 1$$

For example, $f(1) = f(0) + f(0) + 1 = 3$, $f(2) = f(1) + f(1) + 1 = 7$, etc. A possible approach to solve this problem is to develop an algorithm that computes f recursively, using the given two equations directly. Provide your opinion whether the algorithm is efficient or not. If it is not efficient, describe a way in which the algorithm can be improved.

No this algorithm is not efficient enough. To make it more efficient we can keep values in an array or list. Rather than solving it recursively at every step