Ollscoil Mhá Nuad
Maynooth University

SEMESTER 2
2023-2024

CS211FZ
Algorithms and Data Structures 2

Dr. Irfan Ahmad, Dr. Mahwish Kundi, Dr. Michael English

Time allowed: 2 hours.

Answer at least three questions.
Your mark will be based on your best three answers.
All questions carry equal marks.

Instructions

	Yes	No
Log Books Allowed		X
Formula Tables Allowed		X
Dictionary allowed (supplied by the student)		X
Non-programmable calculator allowed	X	
Students required to write in and return the exam question paper		X

QUESTION 1 [25 Marks]

To analyze the INSERTION-SORT algorithm, the time cost of each statement and the number of times each statement is executed are given below in Figure 1.

a) Calculate the total running time for the Insertion-Sort algorithm. [10 marks]

b) What is the best case for the Insertion-Sort algorithm? Calculate the running time for the best case and write the answer in big O notation. [5 marks]

c) What is the worst case for the Insertion-Sort algorithm? Calculate the running time for the worst case and write the answer in big O notation. [5 marks]

d) Explain why we usually concentrate on finding the worst-case running time. [5 marks]

INSERTION-SORT(A, n)	*cost*	*times*
1 **for** $i = 2$ **to** n	c_1	n
2 $key = A[i]$	c_2	$n-1$
3 // Insert $A[i]$ into the sorted subarray $A[1:i-1]$.	0	$n-1$
4 $j = i-1$	c_4	$n-1$
5 **while** $j > 0$ and $A[j] > key$	c_5	$\sum_{i=2}^{n} t_i$
6 $A[j+1] = A[j]$	c_6	$\sum_{i=2}^{n}(t_i - 1)$
7 $j = j-1$	c_7	$\sum_{i=2}^{n}(t_i - 1)$
8 $A[j+1] = key$	c_8	$n-1$

Figure 1. Pseudocode for Insertion-Sort algorithm

QUESTION 2 [25 Marks]

The data stored in the hash table is shown below in Figure 2.

a) Calculate the load factor of the hash table. [5 marks]

b) Given that the load factor is quite high, what would you recommend to reduce the number of collisions? [5 marks]

c) Show how the keys 583, 629 and 975 would be inserted using Quadratic probing. [15 marks]

Hint: Use the hash function h(k) = k % m, where **k** is the key and **m** is the size of the table.

0	995	**8**	152	**16**		**24**		**32**	
1	38	**9**		**17**	535	**25**		**33**	662
2	704	**10**	158	**18**	943	**26**		**34**	737
3	743	**11**	566	**19**	611	**27**	730	**35**	886
4	596	**12**	974	**20**	390	**28**	398	**36**	73
5	403	**13**	626	**21**		**29**	732		
6	931	**14**	532	**22**		**30**	881		
7	426	**15**	52	**23**	652	**31**	364		

Figure 2. Hash Table

QUESTION 3 [25 Marks]

The graph represented by the adjacency matrix is shown below in Figure 3.

a) Draw the undirected unweighted graph represented by the adjacency matrix below. [10 marks]

b) Show how a breadth-first search would traverse the graph starting at vertex **s**. [10 marks]

c) What is the time complexity of the breadth-first search algorithm? [5 marks]

	r	s	t	u	v	w	x	y	z
r	0	1	1	0	0	1	0	0	0
s	1	0	0	1	1	0	0	0	0
t	1	0	0	1	0	0	0	0	0
u	0	1	1	0	0	0	0	1	0
v	0	1	0	0	0	1	0	1	0
w	1	0	0	0	1	0	1	0	1
x	0	0	0	0	0	1	0	1	1
y	0	0	0	1	1	0	1	0	0
z	0	0	0	0	0	1	1	0	0

Figure 3. Adjacency Matrix of an Unweighted, Undirected Graph

QUESTION 4 [25 Marks]

The frequency of characters is shown below in Figure 4.

a) Draw a Huffman code tree based on the frequency of characters. [15 marks]

b) Assign codes to each character. [5 marks]

c) Calculate the resulting Huffman compression ratio, assuming an original ASCII encoding of 3 bits per character. [5 marks]

Characters	**Frequencies**
A	45
B	13
C	12
D	16
E	9
F	5
G	3

Figure 4. Frequency of characters