# UNIVERSITI MALAYSIA PERLIS

Ujian
Sidang Akademik 2021/2022

Disember 2021

## NMJ 20003 – ALGORITHM AND DATA STRUCTURES
## [Algorithma dan Struktur-struktur Data]

Masa : 1 jam

Please make sure that this question paper has **SEVEN (7)** printed pages including this front page before you start the examination.
*[Sila pastikan kertas soalan ini mengandungi **TUJUH (7)** muka surat yang bercetak termasuk muka hadapan sebelum anda memulakan peperiksaan ini.]*

This question paper has **FOUR (4)** questions with total marks of 40. Answer **ALL** questions.
*[Kertas soalan ini mengandungi* **EMPAT (4)** *soalan dengan jumlah markah 40. Jawab* ***SEMUA*** *soalan dari bahagian ini.]*

| | |
|---|---|
| Name | |
| Matric No | |
| Mark | Question 1:<br>Question 2:<br>Question 3:<br>Question 4:<br><br>Total : |

**Question 1 (10 Marks)**

(CO1, PO2, C2)

(a) The performance of algorithms can be measured on the scales of time and space. Differentiate between time and space to measure the performance.

(2 Marks/Markah)

(CO1, PO2, C3)

(b) Calculate the frequency count for each line of the code segment in Table 1.1. Write the answer in the Frequency column.

**Table 1.1**

| No. | Code Segment | Frequency |
|---|---|---|
| 1 | for (int i = 0; i < n; i++) | |
| 2 | a = a + rand(); | |
| 3 | End for | |
| 4 | int j = 1; | |
| 5 | while (j < n) | |
| 6 | j *= 2; | |
| 7 | End while | |

(6 Marks/Markah)

(CO1, PO2, C3)

Based on Table 1.1,

i. Calculate the total frequency count.

(1 Mark/Markah)

ii. Define the “Big O” notation.

(1 Mark/Markah)

**Question 2**

(CO1, PO2, C3)

(a) Calculate the computational complexity (in the "Big-O" sense) of the following piece of code in Figure 2.1.

```
1    int n = 1000;
2    int[] x = new int[n];
3    int sum = 0;
4    for (int j = 0; j < n; j++)
5        for (int k = 0; k < n; k++)
6            sum += j * k;
7        End
8    End
```

**Figure 2.1**

(4 Marks/*Markah*)

(CO2, PO1, C3)

(b) The number of array, Student is given by *(u1-L1+1) (u2-L2+1)*. Find the number of elements in Student [-1:3 , 2:5].

(2 Marks/*Markah*)

(CO2, PO1, C3)

(c) Find the addresses of the element Vehicle[-1,-4,2], in the array, Vehicle[-2:4, -6:10,1:3] with base address $\alpha = 200$. The address of array Vehicle [*i* , *j*, *k*] is given by $\alpha + (i - L1)(u2 - L2 + 1)(u3 - L3 + 1) + (j - L2)(u3 - L3 + 1) + (k - L3)$.

(2 Marks/*Markah*)

(CO2, PO1, C3)

(d) A sparse matrix is a matrix with zeros as the dominating elements. Obtain an array representation for the sparse matrix in Figure 2.2.

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

**Figure 2.2**

(2 Marks/*Markah*)

**Question 3**

(CO2, PO1, C3)
Consider the infix expression below.

(a) Convert the infix expression to postfix expression without stack.

*+ (a * b / c) - d ^ e – f * g*

(2 Marks/*Markah*)

(b) Convert the infix to postfix expression for Question 3(a) using stack data structure.

| Symbol | Stack | Postfix |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

(8 Marks/*Markah*)

**Question 4**

(CO2, PO1, C2)

(a) Define a circular queue data structure.

(2 Marks/*Markah*)

(CO2, PO1, C3)

(b) Consider a single array Q[1:5]. Initially Q[1:5] is empty and all variables front and rear are initialized to zero (0) as shown in Figure 4.1. Given the Circular Queue operation functions are described in Figure 4.2 and a pseudocode in Table 4.1, demonstrate the CIRCULAR QUEUE operation and the output of the statements. Use Table 4.2 in your answer.

(8 Marks/*Markah*)

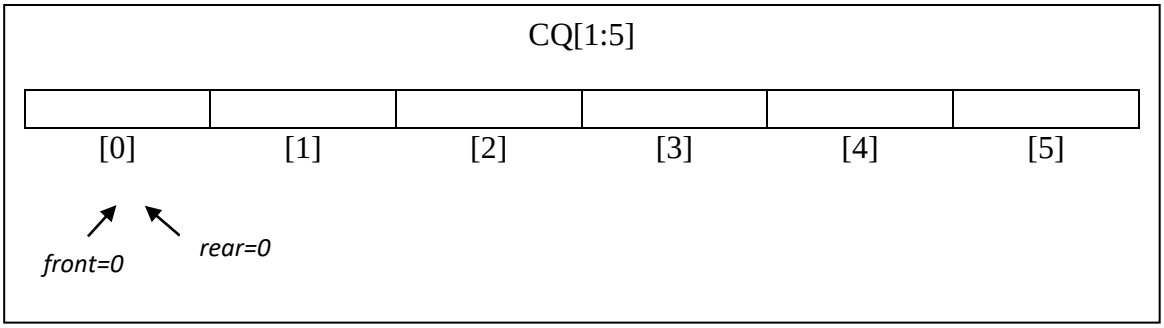


**Figure 4.1**

**ENQ(Q,DATA) : Insert DATA value into Circular Queue Q.**
**DEQ(Q,DATA) : Delete a value from Circular Queue Q and assign it to DATA**
**PRINT(DATA) : Display the value of DATA**
**EMPTYQ(Q) : Is a Boolean function which returns true if Q is empty and false otherwise**

**Figure 4.2**

**Table 4.1**

| Line | Statement |
|---|---|
| 1 | DATA: = 1 |
| 2 | Do |
| 3 | If DATA < 10 Then |
| 4 | ENQ(Q,DATA) |
| 5 | DATA:=2 * DATA |
| 6 | Else |
| 7 | DEQ(Q,DATA) |
| 8 | PRINT(DATA) |
| 9 | DATA:=2 * DATA + 1 |
| 10 | WHILE (not EMPTYQ(Q) && DATA<30) |
| 11 | PRINT (DATA) |
| 12 | END |

**Table 4.2**

| Line No | Queue | Rear | Front | Item | Output |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

**-oooOooo-**