UNIVERSITI MALAYSIA PERLIS

Peperiksaan Akhir Semester Pertama
Sidang Akademik 2016/2017

Disember 2016 / Januari 2017

EKT 224 - Algorithm and Data Structure
[Algoritma dan Struktur Data]

Masa : 3 jam

Please make sure that this question paper has **THIRTEEN (13)** printed pages including this front page before you start the examination.
*[Sila pastikan kertas soalan ini mengandungi **TIGA BELAS (13)** muka surat yang bercetak termasuk muka hadapan sebelum anda memulakan peperiksaan ini.]*

This question paper has **TWO (2)** sections. **Section A** consists of **THREE (3)** questions. **Section B** consists of **THREE (3)** questions. Answer **ALL** questions in **Section A** and answer only **TWO (2)** questions in **Section B**. Each question contributes 20 marks.
*[Kertas soalan ini mengandungi **DUA (2)** bahagian. **Bahagian A** mengandungi **TIGA (3)** soalan dan **Bahagian B** mengandungi **TIGA (3)** soalan. Jawab **SEMUA** soalan pada **Bahagian A** dan jawab hanya **DUA (2)** soalan pada **Bahagian B**. Markah bagi tiap-tiap soalan adalah 20 markah.]*

Dicetak oleh Unit Peperiksaan dan Pengijazahan, Bahagian Pengurusan Akademik, Jabatan Pendaftar

SECTION A
[BAHAGIAN A]

Answer **ALL** questions under this section.
*[Jawab **SEMUA** soalan di dalam bahagian ini.]*

QUESTION 1
[SOALAN 1]

(C1, CO1, PO3)

(a) Describe apriori complexity analysis of an algorithm in the estimation of the computation efficiency.
[Terangkan algoritma kerumitan apriori untuk menganggarkan kecekapan pengiraan.]

(4 Marks/*Markah*)

(C2, CO1, PO3)

(b) **Table 1** shows a fragment of pseudo code.
*[**Jadual 1** menunjukkan keratan kod sudo.]*

(i) Compute the frequency of each statement.
[Kirakan kekerapan setiap kenyataan.]

(6 Marks/*Markah*)

(ii) Determine the total frequency into time complexity by using Big-O notation.
[Tentukan jumlah kekerapan kepada masa kerumitan menggunakan notasi Big-O.]

(2 Marks/*Markah*)

Table 1
[Jadual 1]

Line No	Statements	Frequency
1	Int I, J, K, N;	
2	For (I = 11; I<= 20; I++) {	
3	For (J =1; J<=N; J++){	
4	For (K =1; K<=N-1; K++){	
5	For (L=0; L< 10; L++){	
6	Print("UniMAP"); }}}}	
	Total	

(C4, CO1, PO3)

(c) Consider $T_1(n)$, $T_2(n)$, and $T_3(n)$ are the time complexities of three program fragments P1, P2 and P3 where $T_1(n) = O(f(n))$, $T_2(n) = O(g(n))$ and $T_3(n) = O(h(n))$.
[Anggapkan $T_1(n)$, $T_2(n)$, dan $T_3(n)$ adalah kerumitan masa tiga keratan aturcara P1, P2 dan P3 di mana $T_1(n) = O(f(n))$, $T_2(n) = O(g(n))$ dan $T_3(n) = O(h(n))$.]

Solve the time complexity for $T_1(n) * T_2(n) * T_3(n)$.
[Selesaikan kerumitan masa bagi $T_1(n) * T_2(n) * T_3(n)$]

(2 Marks/*Markah*)

(C4, CO1, PO3)

(d) A recurrence relation for a recursive procedure is shown as below:
[Satu hubungan pengulangan untuk suatu prosidur rekursi adalah seperti berikut:]

$T(N) = a$ $N<=0;$
$T(N) = 2T(N - 1) + b$ $N > 0;$

(6 Marks/*Markah*)

Solve the recurrence relation and prove the time complexity of the procedure is $O(2^N)$.
[Selesaikan hubungan pengulangan tersebut dan buktikan masa kerumitan bagi prosidur tersebut adalah $O(2^N)$]

QUESTION 2
[SOALAN 2]

(C2, CO1, PO3)

(a) A 3-dimensional array is given as A[1:4, 1:3, 1:2]. Each element of this array occupies 2-bytes in the memory. If the base address of A is 22000, calculate the address of the element in the index A[2,1,2].
[Satu 3-dimensi tatasusunan diberikan sebagai A [1:4, 1:3, 1:2]. Setiap elemen tatasusunan ini menggunakan 2-bait dalam ingatan. Jika alamat asas dari A adalah 22000, hitungkan alamat unsur dalam indeks A [2,1,2]]

(4 Marks/*Markah*)

(C3, C5, CO2, PO3)

(b) Given the two (2) Sparse Matrices in **Figure 2.1**.
*[Diberikan dua (2) matrik jarang di dalam **Rajah 2.1**.]*

8	0	0	6	0	9
0	2	0	0	3	0
0	5	0	0	0	0
0	0	3	0	0	0

A[1:4, 1:6]

(a)

1	0	0	9	0
0	3	0	0	0
0	6	0	0	0
0	0	0	0	7

B[1:4, 1:5]

(b)

Figure 2.1
[Rajah 2.1]

(i) Construct the compact representation of the two (2) sparse matrices.
[Bagunkan perwakilan padat bagi dua (2) matrik jarang tersebut].

(4 Marks/*Markah*)

(ii) Suppose, for Matrix A, each element size is set **2-bytes** and for Matrix B, each element size is set to **3-bytes.** Analyze the efficiency of the compact representation of Matrix A and B.
[Andaikan Matrik A mempunyai saiz 2-bait saiz dan Matrik B mempunyai saiz 3-bait untuk setiap elemen. Lakukan analisa keberkesanan bagi setiap perwakilan padat pada Matrik A dan B.]

(2 Marks/*Markah*)

...5/13

(C3, CO2, PO3)

(c) Illustrate step by step the logical representation of the pseudo code in the **Figure 2.2**, where T is a singly linked list. Use the link list T in **Figure 2.3** in your answer.
*[Gambarkan langkah demi langkah perwakilan logik kod sudo di **Rajah 2.2**, di mana T adalah senarai satu pautan. Gunakan senarai berpaut T pada **Rajah 2.3** di dalam jawapan anda.]*

(5 Marks/*Markah*)

```
Procedure DELETE_LAST(T)
  IF (T = = NIL) THEN
     Call ABANDON_DELETE;
  ELSE
  {
     TEMP = T;
     WHILE (LINK(TEMP) ≠ NIL)
     {
        PREVIOUS_PTR = TEMP
        TEMP = LINK(TEMP)
     }
     LINK(PREVIOUS_PTR) = NIL;
     Call RETURN(TEMP);
}
END DELETE_LAST.
```

Figure 2.2
[Rajah 2.2]

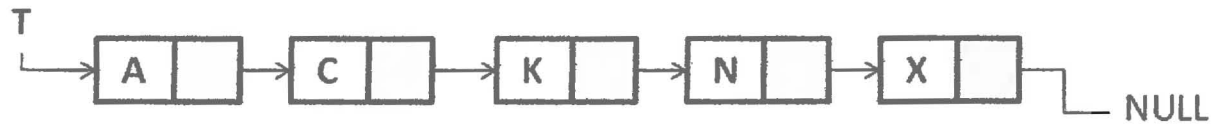

Figure 2.3
[Rajah 2.3]

...6/13

(C3, CO2, PO3)

(d) Given a memory snap shot in the **Table 2.1**, where **START** and **AV_SP** store the start pointers of the Linked List and the Available Space List respectively.
*[Diberi satu gambaran ingatan di dalam **Jadual 2.1**, di mana **START** dan **AV_SP** menyimpan petunjuk mula bagi satu Senarai Berpaut dan Senarai Ruang Tersedia.]*

Table 2.1
[Jadual 2.1]

START: 2 **AV_SP: 4**

	DATA	LINK
1	11	0
2	17	6
3	24	9
4	94	5
5	34	7
6	87	8
7	34	1
8	46	3
9	19	10
10	56	0

(i) Construct the Linked List and Available Space List.
[Bangunkan Senarai Berpaut dan Senarai Ruang Tersedia.]

(2 Marks/*Markah*)

(ii) Illustrate how the Linked List and Available Space List are affected when delete data 24 operation is carried out. Conclude your answer.
[Gambarkan bagaimana Senarai Berpaut dan Senarai Ruang Tersedia terkesan akibat dari operasi pembuangan data 24 dilakukan. Berikan kesimpulan jawapan anda.]

(3 Marks/*Markah*)

QUESTION 3
[SOALAN 3]

(C2,CO2, PO3)

(a) For the following logical expression (A and B and C) or D or (not H).
[Untuk ekspresi logikal berikut (A dan B dan C) atau D atau (tidak H).]

(i) Obtain the equivalent postfix and prefix expression.
[Dapatkan ekspresi "postfix" dan "prefix".]

(3 Marks/*Markah*)

(ii) Evaluate the postfix expression for A=*true*, B=*false*, C=*true*, D=*true*, E=*true*, H=*false*.
[Nilaikan ekspresi "postfix" untuk A=benar, B=salah, C=benar, E=benar, H=salah.]

(5 Marks/*Markah*)

(C3,CO2,PO3)

(b) With the aid of STACK operation using Linked List data structure trace on the following expression to check whether parentheses are balanced:
[Dengan bantuan operasi STACK menggunakan struktur data senarai berpaut, surih ekspresi berikut untuk menentukan samada kurungan adalah seimbang.]

$$((X + Y) * Z / (M - N)$$

(6 Marks/*Markah*)

(C2,CO2,PO3)

(c) DEQ[1:7] is an example of output restricted deque implemented as a *circular array* and LEFT and RIGHT indicate the ends of the deque as shown in **Figure 3.1**. INSERT ('xx', [LEFT|RIGHT]) indicates the insertion of the data item at the left or right end, and DELETE() deletes the item from the left end only.
[DEQ[1:7] adalah satu contoh keluaran terhad 'deque' yang dilaksanakan sebagai tatasusunan pekeliling dimana KIRI dan KANAN menunjukkan hujung 'deque' seperti yang ditunjukkan di dalam ***Rajah 3.1****. INSERT (' xx ' , [KIRI | KANAN]) menunjukkan operasi kemasukan maklumat di hujung kiri atau kanan, dan DELETE () operasi membuang maklumat dari hujung sebelah kiri sahaja.]*

DEQ

[1]	[2]	[3]	[4]	[5]	[6]	[7]
	A	B	C	D		

Figure 3.1
[Rajah 3.1]

...8/13

Execute the following insertions and deletions on DEQ. At each execution, show the position of LEFT and RIGHT:
[Laksanakan sisipan dan potongan berikut pada DEQ. Pada setiap pelaksanaan, sila tunjukkan kedudukan KIRI dan KANAN :]

(i)	INSERT('E', LEFT)	(1 Mark/*Markah)*
(ii)	INSERT('F', RIGHT)	(1 Mark/*Markah*)
(iii)	DELETE	(1 Mark/*Markah*)
(iv)	INSERT('G', LEFT)	(1 Mark/*Markah*)
(v)	INSERT('H', LEFT)	(1 Mark/*Markah*)
(vi)	INSERT('I', RIGHT)	(1 Mark/*Markah*)

SECTION B
Answer **ONLY TWO** questions from this section.
*[Jawab **DUA (2)** soalan dari bahagian ini.]*

QUESTION 4
[SOALAN 4]

(C5, CO2, PO3)

(a) Consider a binary tree T with 9 nodes. The Inorder (LPR) and Preorder (PLR) traversal of T yield the following:
[Pertimbangkan pepohon binari T yang mempunyai 9 nod. Dalam urutan (LPR) dan Pra urutan (PLR) T menghasilkan berikut:]

Inorder (I) : E A C K F H D B G
Preorder (P) : F A E K C D H G B

(i) Design the binary tree.
[Rekabentuk pepohon binari.]

(4 Marks/*Markah*)

(ii) Obtain the array and linked representation of the binary tree.
[Hasilkan tatasusunan dan senarai berpaut bagi pepohon binari di atas.]

(4 Marks/*Markah*)

(iii) Obtain the post order traversal (LRP) of the binary tree.
[Hasilkan laluan urutan terdahulu (LPR) bagi pepohon binari tersebut.]

(2 Marks/*Markah*)

(C4, CO2, PO3)

(b) Analysed the undirected and weighted graph in **Figure 4.1**.
*[Analisa graf tak terarah dan wajaran dalam **Rajah 4.1.**]*

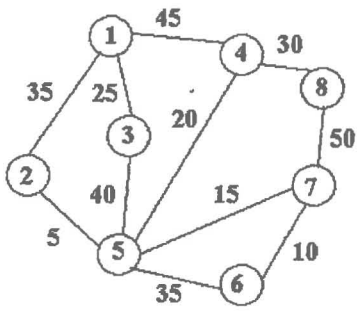

Figure 4.1
[Rajah 4.1]

(i) Extract minimum cost spanning tree starting from the node 2.
[Ekstrak kos pepohon rangkuman minima bermula dari nod 2.]

(5 Marks/*Markah*)

(ii) Traverse the graph using **breadth first traverse** algorithm starting from the node 2.
*[Rentas graf menggunakan **'breadth first search'** bermula nod 2.]*

(5 Marks/*Markah*)

Hints: You should use the following table to represent your answer.
[Petua: Anda perlu menggunakan jadual berikut untuk memasukkan jawapan anda]

Current Vertex	Queue Q[1:10]	Traversal Output

QUESTION 5
[SOALAN 5]

(C5, CO3, PO3)

(a) Consider a set of keys in set S. The hash function is **h(X) = X mod 9** and assume that each bucket has three (3) slot.
*[Pertimbangkan satu set kekunci di dalam set X. Fungsi 'hash' adalah **h(X) = X mod 9** dan andaikan setiap takungan mempunyai tiga (3) slot]*

X = {16, 8, 35, 55, 10, 3, 70, 85, 54, 9, 38, 48, 51, 81, 30, 12, 21, 34, 43, 19, 59, 27}

(i) Construct a hash table H[0:8, 0:2] using ***quadratic probing*** method to insert the keys in set X.
*[Bina jadual hash H[0:8, 0:2] menggunakan kaedah '**quadratic probing**' untuk menyelit kekunci ke dalam set X.]*

(7 Marks/*Markah*)

(ii) Suggest another efficient method if all the keys are not fit in the hash table.
[Cadangkan satu kaedah efisien yang lain sekiranya kesemua kekunci tidak muat di dalam jadual 'hash'.]

(3 Marks/*Markah*)

(C4, CO3, PO3)

(b) Write Bubble sort algorithm to sort some unordered data into a descending order form. With the aid of the sorting algorithm sort the unordered list below, into a descending ordered form.
[Tulis algoritma susunan Buih untuk menyusun beberapa data tidak tertib ke dalam bentuk susunan menurun. Dengan bantuan algoritma isihan, susunkan senarai tidak tertib di bawah kepada susunan menurun.]

{10, 13, 2, 16, 32, 44, 23, 14, 43, 50, 34, 33, 1, 5, 100}

(10 Marks/*Markah*)

QUESTION 6
[SOALAN 6]

(C5, CO2, PO4)

(a) Consider the list of number below.
[Pertimbangkan senarai nombor di bawah.]

(i) Apply quick sort algorithm to rearrange the following numbers into ascending order. Indicate clearly the pivots that you use.
[Aplikasikan algoritma isihan cepat bagi menyusun secara menaik senarai nombor-nombor di bawah. Tunjukkan penggunaan 'pivot' yang digunakan.]

{220 218 223 112 107 126 119 316 124 140}

(6 Mark/*Markah*)

(ii) Explain why quick sort is not a stable sort.
[Terangkan kenapa isihan cepat merupakan isihan yang tidak stabil]

(2 Marks/*Markah*)

(iii) State the worst time complexity for quick sort and explain its condition.
[Nyatakan kerumitan masa terburuk bagi isihan cepat dan terangkan situasinya.]

(2 Marks/*Markah*)

(C4, CO3, PO3)

(b) Consider the given sorted Data List as follows. Here the n (n=20) the number of elements is such that $F_9 > (n+1)$ and $F_8 + m = (n+1)$, where m=0 and n=20.
[Pertimbangkan Senarai Data terisih dan Senarai Carian yang diberikan. Nilai n (n=20), nombor element adalah $F_9 > (n+1)$ dan $F_8 + m = (n + 1)$ di mana, $m = 0$ dan $n = 20$.]

Data List: **{2, 4, 8, 9, 17, 36, 44, 55, 81, 94, 116, 221, 256, 302, 356, 396, 401, 434, 536}**

Search List: **{434, 66}**

(i) Design algorithm when the *Fibonacci search* is undertaken on the keys belonging to the search list.
[Reka bentuk algoritma apabila carian 'Fibonacci' dilakukan ke atas kekunci yang dimiliki oleh senarai carian.]

(5 Marks/*Markah*)

(ii) Illustrate the algorithm in (i) by using the given Data List and Search List.
[Gambarkan algoritma di dalam (i) dengan menggunakan Senarai Data dan Senarai Carian yang diberikan.]

(5 Marks/*Markah*)

-oooOooo-

Course Outcomes (COs)

CO1	Ability to analyze algorithms and to determine algorithm correctness and time efficiency.
CO2	Ability to design and implement linear and non linear data structures
CO3	Ability to design and implement searching and sorting algorithms

Program Outcomes (POs)

PO 01	Ability to acquire and apply knowledge of mathematics, science, engineering and an in-depth technical competence in computer engineering discipline to solve the complex engineering problem
PO 02	Ability to identify, formulate and solve complex engineering problems.
PO 03	Ability to design solutions for complex engineering problems and systems, components or processes to meet desired needs.
PO 04	Ability to conduct investigation into complex problems as well as to analyze and interpret data.
PO 05	Ability to use techniques, skills and modern engineering tools necessary for complex engineering practices so as to be easily adaptable to industrial needs.
PO 06	Understanding of the social, cultural, global and environmental responsibilities of a professional engineer.
PO 07	Ability to have entrepreneurship, the process of innovation and the need for environmental and sustainable development.
PO 08	Ability to understand the professional and ethical responsibilities and commitment to the community.
PO 09	Ability to function on multi-disciplinary teams.
PO 10	Ability to communicate effectively on complex engineering activities with the engineering community and with society at large
PO 11	A Recognition of the need for, and an ability to engage in life-long learning
PO 12	Demonstrate the understanding of project management and finance principles