UNIVERSITI MALAYSIA PERLIS

Peperiksaan Semester Pertama
Sidang Akademik 2014/2015

8 Januari 2015

EKT 224 – Algorithm and Data Structures
Algoritma dan Struktur- Struktur Data

Duration: 3 Hours
Masa: 3 jam

Please make sure that this question paper has **TEN (10)** printed pages including this front page before you start the examination.
*[Sila pastikan kertas soalan ini mengandungi **SEPULUH (10)** muka surat yang bercetak termasuk muka hadapan sebelum anda memulakan peperiksaan ini.*]

This question paper has **FIVE (5)** questions. Answer **ALL** questions. Each question contributes 20 marks.
*[Kertas soalan ini mengandungi **LIMA (5)** soalan. Jawab **SEMUA** soalan. Markah bagi tiap-tiap soalan adalah 20 markah.]*

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

QUESTION 1
[SOALAN 1]

[C3, CO1, PO2]

a) Consider the following recurrence relation where T(n) is the time complexity of some statements.
[Pertimbangkan hubungan berulang berikut di mana T (n) adalah kerumitan masa bagi beberapa kenyataan.]

$$T(n) = 2 * T(n/2) + c * n; \quad \text{if } n \geq 2$$
$$= d ; \quad \text{if } n < 2$$

(i) Demonstrate the generic solution for time T(n) at k^{th} steps using substitution method.
[Tunjukkan penyelesaian umum untuk masa T (n) pada langkah ke-k dengan menggunakan kaedah penggantian.]

[5 Marks/*Markah*]

(ii) Solve the k^{th} step for T(n) when $n=2^k$.
[Selesaikan langkah ke-k untuk T (n) apabila $n = 2^k$.]

[3 Marks/*Markah*]

[C4, CO1, PO2]

b) Consider the following fragment of C code in Table 1.
[Pertimbangkan serpihan kod C dalam Jadual 1 berikut.]

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

[4 Marks/*Markah*]

Table 1
[Jadual 1]

Line No	Statements	Frequency
1	For I = 2 to n do	
2	Key1 = 5;	
3	Key2 = 3;	
4	For J = (I – 1) to 1 do	
5	Key3 = Key1 + Key2;	
6	End For	
7	Key3 = Key3 + 1;	
8	End For	
	Total	

(ii) Discriminate the total frequency into time complexity by using Big-O notation.

[Dapatkan jumlah kekerapan ke dalam kerumitan masa dengan menggunakan Big-O notasi.]

[3 Marks/*Markah*]

[C4, CO1, PO2]

c) Suppose a 3-dimentional 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 0xABCD, calculate the address of the element in the index A[2,1,3].

[Anggapkan tatasusunan 3-dimensi diberikan sebagai A [1: 4, 1: 3, 1: 2]. Setiap elemen tatasusunan ini menggunakan 2-bait dalam ingatan. Jika alamat asas dari A adalah 0xABCD, hitungkan alamat unsur dalam indeks A [2,1,3].]

[5 Marks/*Markah*]

QUESTION 2
[SOALAN 2]

Table 2 is a pseudo code of a series of operations on a stack S. PUSH(S, X) pushes an element X into S, POP(S, X) pops out an element from stack S as X, PRINT (X) displays the variable X and EMPTYSTACK (S) is a Boolean function which returns true if S is empty and false otherwise.
[Jadual 2 adalah kod pseudo dari serangkaian operasi pada tumpukan S. PUSH (S, X) menolak unsur X ke dalam tumpukan S, POP (S, X) menarik keluar unsur dari tumpukan S sebagai X, PRINT (X) mempamerkan pemboleh ubah X dan EMPTYSTACK (S) adalah fungsi Boolean yang mengembalikan nilai benar jika S adalah kosong dan palsu jika sebaliknya.]

[C3, CO2, PO3]

(a) Demonstrate the output from the Table 2.
[Tunjukkan keluaran dari Jadual 2.]

Table 2
[Jadual 2]

Line No.	Statement	Line No.	Statement
1	X := 10;	9	PUSH(S, Z);
2	Y := 20;	10	POP(S, X);
3	Z := 30;	11	PUSH(S, 20);
4	PUSH(S, X);	12	PUSH(S, X);
5	PUSH(S, 40);	13	**WHILE NOT** EMPTYSTACK (S) **DO**
6	POP (S, Z);	14	POP (S, X);
7	PUSH(S, Y);	15	PRINT(X);
8	PUSH(S, 30);	16	**End WHILE**

[10 Marks/*Markah*]

[C3, CO2, PO3]

(b) Re-write lines 13 to 16 if only the bottom element of the stack S need to be printed.
[Tuliskan kembali baris 13 hingga 16 hanya jika elemen bahagian bawah tumpukan S diperlukan untuk dicetak.]

[4 Marks/*Markah*]

[C3, CO2, PO3]

(c) Re-write lines 13 to 16 to keep the elements of the stack S unchanged.
[Tuliskan kembali baris 13 hingga 16 untuk menjaga unsur dari tumpukan S supaya tidak berubah.]

[6 Marks/*Markah*]

QUESTION 3
[SOALAN 3]

[C4, CO2, PO3]

(a) Figure 1 (a) and Figure 1 (b) show logical representation of deletion in a singly and circularly linked list respectively. Criticize the advantages of delete operation for the Figure 1 (b) over Figure 1 (a).
[Rajah 1 (a) dan Rajah 1 (b) menunjukkan perwakilan logik penghapusan dalam senarai secara tunggal dan senarai linkungan berkait masing-masing. Kritik kelebihan operasi penghapusan bagi Rajah 1 (b) di atas Rajah 1 (a).]

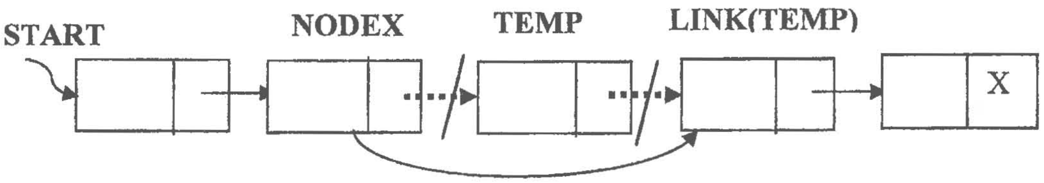

Figure 1 (a)
[Rajah 1 (a)]

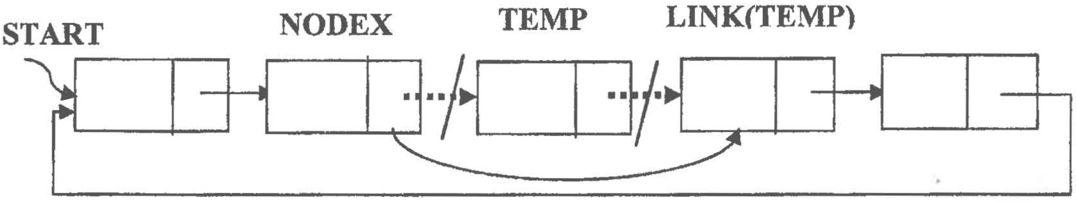

Figure 1 (b)
[Rajah 1 (b)]

[5 Marks/ *Markah*]

[C3, CO2, PO3]

(b) Figure 2 shows a headed circular doubly linked list called HCDLL. LLINK(Y) indicates Left node of Y and RLINK(Y) indicates Right node of Y. Sketch and write algorithm for the insertion operation of node X into the HCDLL after node Y.
[Rajah 2 menunjukkan senarai pautanberganda yang berlingkeryang berkepadadisebut sebagai HCDLL. LLINK (Y) menunjukkan nod Kiri Y dan RLINK (Y) menunjukkan nod kanan Y. Lakar dan tuliskan algoritma untuk operasi kemasukkan nod X ke dalam HCDLL selepas nod Y.]

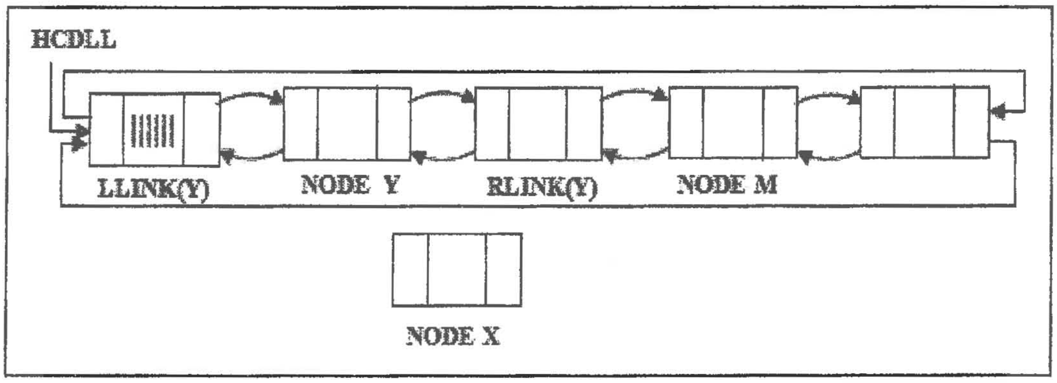

Figure 2
[Rajah: 2]

[5 Marks/ *Markah*]

[C3, CO2, PO3]

(c) Consider the following infix expression.
[Pertimbangkan ungkapan infiks berikut.]

$$(x + y + z) \wedge (a + b - c) - g * n * m + r$$

(i) Design an expression binary tree from the above infix expression.
[Rekabentuk ungkapan pohon binari dari ungkapan infiks di atas.]

[4 Marks/ *Markah*]

(ii) Demonstrate the Inorder Traversal output on the expression binary tree.
[Tunjukkan keluaran penyusuran tidak tersusun atas pokok ungkapan binari.]

[2 Marks/ *Markah*]

(iii) Demonstrate the Preorder Traversal output on the expression binary tree.
[Tunjukkan keluaran penyusuran susunan awalan atas pokok ungkapan binari.]

[2 Marks/ *Markah*]

(iv) Demonstrate the Postorder Traversal output on the expression binary tree.
[Menunjukkan keluaran penyusuran susunan akhiran atas pokok ungkapan binari.]

[2 Marks/ *Markah*]

QUESTION 4
[SOALAN 4]

[C3, CO2, PO3]

(a) Demonstrate and Compute a minimum cost spanning tree (using Prims Algorithm) for the graph in Figure 3.
[Tunjukkan dan buat pengiraan pohon merentang kos yang minimum (menggunakan Prims Algoritma) untuk graf dalam Rajah 3.]

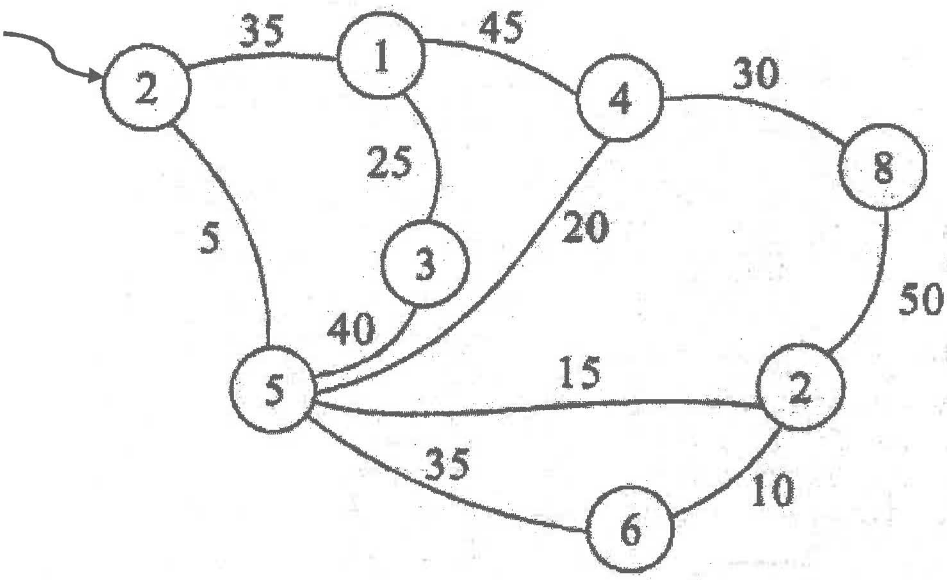

Figure 3
[Rajah 3]

[10 Marks/*Markah*]

Answer Table Prototype:

Edge	Cost of the Edge	V'	E'	Minimum Cost Spanning Tree

[C3, CO3, PO3]

(b) For the set of keys {11, 55, 13, 35, 71, 52, 61, 9, 5, 86, 31, 49, 85, 70} demonstrate the hash table which employing rehashing for collision resolution. Assume the hash table is defined as HT[0…8, 0…2] and consists hashing function H(X) = X *mod* 9 and the rehashing function H`(X) = 7 – (X *mod* 7). The collision resolution function is given by hi = H(X) + i * H`(X) *mod* b, where i= 1, 2, 3, …

[Untuk set kunci {11, 55, 13, 35, 71, 52, 61, 9, 5, 86, 31, 49, 85, 70} tunjukkan jadual hash yang menggunakan ulangan untuk penyelesaian pelanggaran. Andaikan jadual hash ditakrifkan sebagai HT [0 ... 8, 0 ... 2] dan terdiri daripada fungsi hansh H (X) = X mod 9 dan fungsi ulangan H`(X) = 7 - (X mod 7). Fungsi resolusi perlanggaran diberikan oleh hi = H (X) + i * H` (X) b mod, di mana i = 1, 2, 3, ...]

[5 Marks/*Markah*]

[C3, CO3, PO3]

(c) For the same set of keys in question (b) demonstrate the hash table which following quadratic probing for collision resolution. Briefly explain from the observation on the Hash Tables from question (b) and (c).

[Untuk set yang sama kunci dalam soalan (b) tunjukkan jadual hash yang mengikuti penyelesaian kuadratik untuk resolusi perlanggaran. Terangkan secara ringkas dari pemerhatian Jadual Hash dari soalan (b) dan (c).]

[5 Marks/*Markah*]

QUESTION 5
[SOALAN 5]

[C3, CO3, PO3]

(a) Consider a list L={34, 21, 89, 54, 45, 12, 90, 76, 62, 17} of unordered numerical elements. Demonstrate the Transpose sequential Search technique for the list of elements {90, 89, 90, 21, 90, 12 }.
[Pertimbangkan senarai L = {34, 21, 89, 54, 45, 12, 90, 76, 62, 17} untuk unsur-unsur berangka yang tidak tertib. Tunjukkan teknik Ubahan Carian Urutan untuk senarai unsur-unsur {90, 89, 90, 21, 90, 12}.]

[6 Marks/*Markah*]

Answer Table Prototype:

Search Key	List L before search	Number of element comparisons made during the search	List L after search

[C3, CO3, PO3]

(b) Consider a list L = {17, 92, 78, 34, 23, 56, 90, 52, 67, 81, 18}. Demonstrate the Bubble Sort (ascending) technique on the list L. Explain the possibility to obtain worst case time complexity from this sorting algorithm.
[Pertimbangkan senarai L = {17, 92, 78, 34, 23, 56, 90, 52, 67, 81, 18}. Tunjukkan teknik Susunan Gelembung (menaik) dalam senarai L. Jelaskan kemungkinan untuk mendapatkan kes masa kerumitan paling teruk daripada algoritma susunan ini.]

[7 Marks/*Markah*]

[C3, CO3, PO3]

(c) Consider a list L = {17, 92, 78, 34, 23, 56, 90, 52, 67, 81, 18}. Demonstrate the Insertion Sort (descending) technique on the list L. Explain the possibility to obtain worst case time complexity from this sorting algorithm.
[Pertimbangkan senarai L = {17, 92, 78, 34, 23, 56, 90, 52, 67, 81, 18}. Tunjukkan teknik Kemasukkan Susunan (menurun) dalam senarai L. Jelaskan kemungkinan untuk mendapatkan kes masa kerumitan daripada algoritma susunan ini.]

[7 Marks/*Markah*]

-ooOooo-

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