UNIVERSITI MALAYSIA PERLIS

Peperiksaan Semester Pertama
Sidang Akademik 2013/2014

15 Januari 2014

EKT 334 – Algorithm and Data Structures
[Algoritma dan Struktur Struktur Data]

Masa: 3 jam

Please make sure that this question paper has **ELEVEN (11)** printed pages including this front page before you start the examination.
[Sila pastikan kertas soalan ini mengandungi SEBELAS (11) 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]

(C4, CO1, PO3, PO 4, PO11)

(a) Consider the two program segments in the Figure 1.1 (a and b). Analyze the time complexity for the codes using Big *O* notation.

[Pertimbangan dua keratan aturcara de dalam gambarajah 1.1 (a dan b). Analisa kerumitan masa bagi keratan aturcara tersebut dengan menggunakan notasi Big O]

```
for k = 0 to n do
    for i = 0 to n do
        for j = 0 to n do
            print: *;
```

(a)

```
for k = 0 to 10 do
    for i = 0 to n do
        for j = 0 to n do
            print: *;
```

(b)

Figure 1.1
[Rajah 1.1]

[6 Marks/*Markah*]

(C6, CO1, PO3, PO4, PO11)

(b) Figure 1.2 shows the *Selection Sort* algorithm to sort from the smallest to the largest element in a one dimensional array list. The number of comparisons is *(n-1)* for the first iteration, *(n-2)* for the second iteration, and so on. Let *T(n)* denotes the complexity for *Selection Sort* and *c* denote the total number of other operations such as assignments and additional comparisons in each iteration. Estimate the time complexity for this algorithm.

[Rajah 1.2 menunjukkan algoritma bagi isihan pemilihan untuk menyusun elemen terkecil kepada terbesar di dalam senarai tatasusunan satu dimensi. Jumlah perbandingan pada algortma ini adalah (n-1) bagi lelaran pertama, (n-2) bagi lelaran kedua, dan seterusnya. T(n) adalah mewakili kerumitan bagi isihan pemilihan dan c mewakili jumlah bilangan bagi lain-lain operasi seperti pengumpukan dan perbandingan tambahan di dalam setiap lelaran. Kirakan kerumitan masa untuk algoritma ini.]

```
// A[0:n-1] is the array to sort
for  j = 0 to  n-1 do
   iMin = j;
   for  i = j+1 to n-1 do
        if (a[i] < a[iMin])
          iMin = i;
        End if
    End For
    if ( iMin != j )
       swap(a[j], a[iMin]);
   End if
End For
```

Figure 1.2
[Rajah 1.2]

[5 Marks/*Markah*]

(C5, CO1, PO3, PO4, PO11)

(c) Consider following recurrence relation where T(n) is the time complexity of certain statements.

[Pertimbangkan perhubungan berulang berikut di mana T (n) adalah kerumitan masa penyata tertentu.]

$$T(n) = 2, \quad \text{iff } n=2$$
$$= T(n/2) + 3, \quad \text{iff } n > 2$$

(i) Generate the generic solution for time T(n) at i^{th} steps

[Janakan penyelesaian generik untuk masa T (n) pada langkah ke i.]

[3 Marks/*Markah*]

(ii) Generate the solution for T(n) when $n=2^k$ and $i=k-1$.
[Janakan penyelesaian untuk T(n) apabila $n = 2^k$ dan $i = k-1$]

[6 Marks/*Markah*]

QUESTION 2
[SOALAN 2]

(a) Given two (2) sparse matrices in Figure 2.1.
[Diberi dua (2) matrik jarang di dalam rajah 2.1]

A[5:6]

	1	2	3	4	5	6
1	0	1	0	0	0	0
2	0	0	3	0	0	0
3	2	0	0	5	0	0
4	0	0	7	0	0	0
5	0	0	0	0	4	0

(a)

B[5:6]

	1	2	3	4	5	6
1	3	0	0	2	0	0
2	0	0	1	0	0	5
3	4	0	0	8	0	0
4	0	7	0	9	0	0
5	0	3	0	0	0	2

(b)

Figure 2.1
[Rajah 2.1]

(C5, CO2, PO3, PO4, PO11)

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

[4 Marks/*Markah*]

(C4, CO2, PO3, PO4, PO11)

(ii) Suppose each matrix element occupying 2-bytes in memory. Analyze the efficiency of each compact representation in (i).
[Katakan setiap elemen matrik tersenut menggunakan 2-bait ingatan. Lakukan analisa keberkesanan bagi setiap perwakilan padat pada (i).]

[4 Marks/*Markah*]

(C4, CO2, PO3, PO4, PO11)

(iii) If the base address of the matrix A[5:6] is 100 and the matrix B[5:6] presents in sequence in memory (as in Figure 2.2), calculate the address of B[5,6] element.
[Jika alamat asas bagi matriks A[5:6] adalah 100 dan matriks B[5:6] menduduki secara berturutan di dalam ingatan (seperti di dalam rajah 2.2) hitung alamat bagi elemen matriks B[5:6].]

A[5:6]	B[5:6]

Figure 2.2
[Rajah 2.2]

[4 Marks/Markah]

(C5, CO2, PO3, PO4, PO11)

(b) Figure 2.3 shows pictorial view of logical representation of deletion operation in a single linked list called START. Design an algorithm for the Figure 2.3 (b), where the deletion operation needs to be applying on the node TEMP which is to the right of node NODEX (Figure 2.3 (a)).

[Rajah 2.3 menunjukkan pandangan bergambar perwakilan logik bagi operasi **buang** *di dalam satu senarai berpaut START. Rekabentuk satu algorithma bagi rajah 2.3 (b) di mana operasi buang dilakukan keatas nod TEMP yang berada disebelah kanan nod NODEX.]*

[4 Marks/*Markah*]

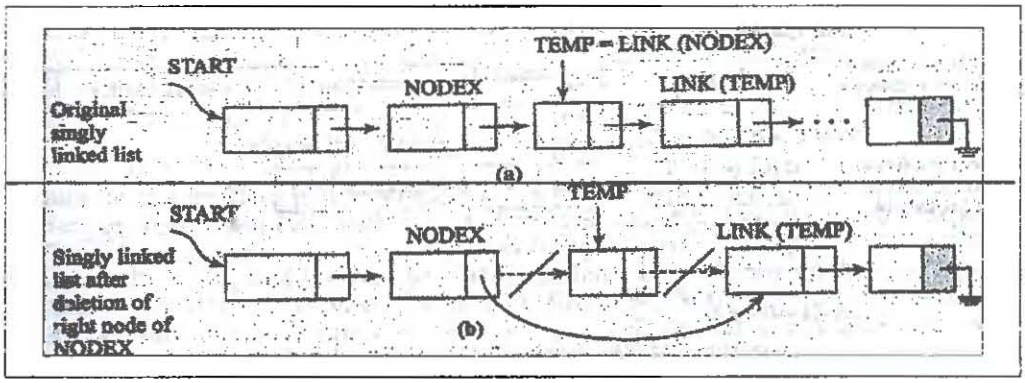

Figure 2.3
[Rajah 2.3]

(C4, CO2, PO3, PO4, PO11)

(c) Figure 2.4 shows doubly linked list T, calculate the missing values in the data fields marked as A and B by using the clues given.

[Rajah 2.4 menunjukkan satu senarai berpaut beganda T, hitungkan nilai-nilai bagi data yang bertanda A dan B dengan menggunakan petunjuk yang diberi.]

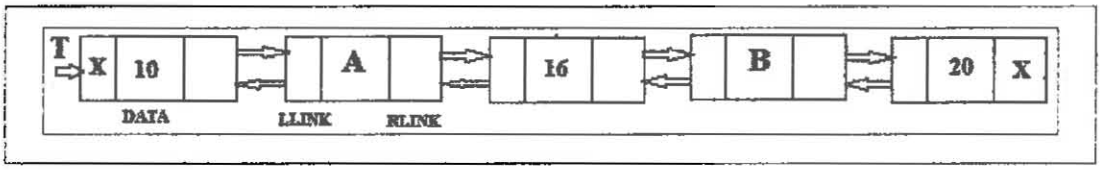

Figure 2.4

[Rajah 2.4]

(i) DATA(A) = DATA (RLINK (RLINK (RLINK(A)))) + 4
+ DATA (LLINK (RLINK (RLINK(A))))
+ DATA (LLINK (LLINK (LLINK (LLINK (RLINK(B)))))).

[2 Marks/*Markah*]

(ii) DATA(B) = DATA(RLINK(LLINK(LLINK(LLINK(RLINK(B)))))) + 5
+DATA(RLINK(LLINK(A))) + DATA(RLINK (RLINK(LLINK(B)))).

[2 Marks/*Markah*]

QUESTION 3
[SOALAN 3]

(C4, CO2, PO3, PO4, PO11

(a) Differentiate between Linear Queues, Deques and Circular Queues.
[Bezakan antara barisan Linear, Deques dan barisan Lengkokan.]

[6 Marks/*Markah*]

(b) Consider a prefix expression [- + ^ / - * A B D C E * * F H I].
*[Pertimbangkan ungkapan awalan [- + ^/- * A B D C E * * F H I]]*

(C5, CO2, PO3, PO4, PO11)

(i) Generate equivalent infix expression.
[Jana ungkapan prasisipan setara.]

[3 Marks/*Markah*]

(C5, CO2, PO3, PO4, PO11)

(ii) Generate equivalent postfix expression.
[Jana ungkapan pascasisipan setara.]

[3 Marks/*Markah*]

(C6, CO2, PO3, PO4, PO11)

(iii) Figure 3.1 shows the procedure to evaluate a postfix expression E using Stack operation. Evaluate the postfix notation in (ii) using Stack operation with A= 2, B= - 2, C= - 4, D= 4, E= 6, F= 16, H= 2, and I= - 2.

[Rajah 3.1 menunjukkan prosedur untuk menilai ungkapan pascasisipan E menggunakan operasi Timbunan . Nilaikan notasi pascasisipan di (ii) menggunakan operasi Timbunan dengan A = 2, B = -2, C = -4, D = 4, E = 6, F = 16, H = 2, dan I = -2.]

[8 Marks/*Markah*]

```
Procedure EVAL_POSTFIX(E)
  X = get_next_character (E); // Get the next character of expression E.
  Case X of
    :X is an operand   : PUSH X into stack S;
    :X is an operator : POP out required number of operands from the Stack
                        STACK [1:5] , evaluate the operator and PUSH the result
                        into the Stack STACK[1:5];
    : X = '$'          : POP out the result from the Stack STACK[1:5] ;
  End Case
End EVAL_POSTFIX(E)
```

Figure 3.1
[Rajah 3.1]

Hints for (iii) Answer Table:
[Petua untuk(iii) Jadual Jawapan:]

PUSH Element X	STACK[1:5]					Action
	[1]	[2]	[3]	[4]	[5]	

QUESTION 4
[SOALAN 4]

(C3, CO2, PO3, PO4, PO11)

(a) Apply the recursive pseudo code in Figure 4.1 on the linked representation of a binary tree in Figure 4.2. Note that for every call of PRINT_NODE(), *START, LC(START)* and *RC(START)* variable need to keep track as given in a table. Here, *LC(START)* and *RC(START)* refer the left child and right child of START node respectively and the *DATA(START)* indicates the element of that node.

[Aplikasikan kod pseudo yang rekursi dalam Rajah 4.1 pada perwakilan rangkaian pepohon binari dalam Rajah 4.2. Perhatikan bahawa bagi setiap panggilan pembolehubah PRINT_NODE (), START, LC (START) dan RC (START) perlu menjejaki seperti yang diberikan dalam jadual. Di sini, LC (START) dan RC (START) merujuk kepada anak kiri dan anak kanan nod START masing-masing dan DATA (START) menunjukkan elemen nod tersebut.]

[7 Marks/*Markah*]

```
1. Procedure PRINT_NODE(START)
   {
2.  If START != NIL then
    {
3.    Call PRINT_NODE( LC(START) );
4.     If ( LC(START) = NIL) &&
                RC(START) = NIL) then
5.          Print DATA(START);
6.    Call PRINT_NODE( RC(START));
    }
}
```

Figure 4.1
[Rajah 4.1]

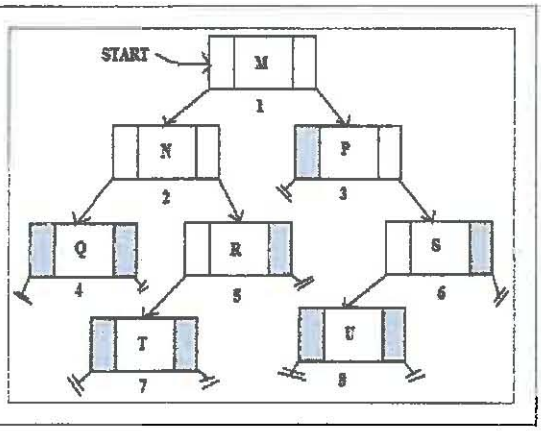

Figure 4.2
[Rajah 4.2]

Answer Table Hints:

Steps	Description	STACK				START	LC(START)	RC(START)

(C5, CO2, PO3, PO4, PO11)

(b) Design a binary tree from the following Inorder and preorder travarsal.
[Reka bentuk pepohon binari dari inorder berikut dan preorder traversal berikut.]

Inorder Traversal : M N O P Q R S T U
Preorder Traversal : Q N M P O S R U T

[5 Marks/*Markah*]

(C3, CO2, PO3, PO4, PO11)

(c) Figure 4.3 shows a weighted directed graph. Apply Dijkstra's algorithm on the Figure 4.3 to find single-source, shortest path.
[Rajah 4.3 menunjukkan graf berwajaran terarah. Aplikasikan algoritma Dijkstra pada Rajah 4.3 untuk mencari satu sumber, jalan paling singkat.]

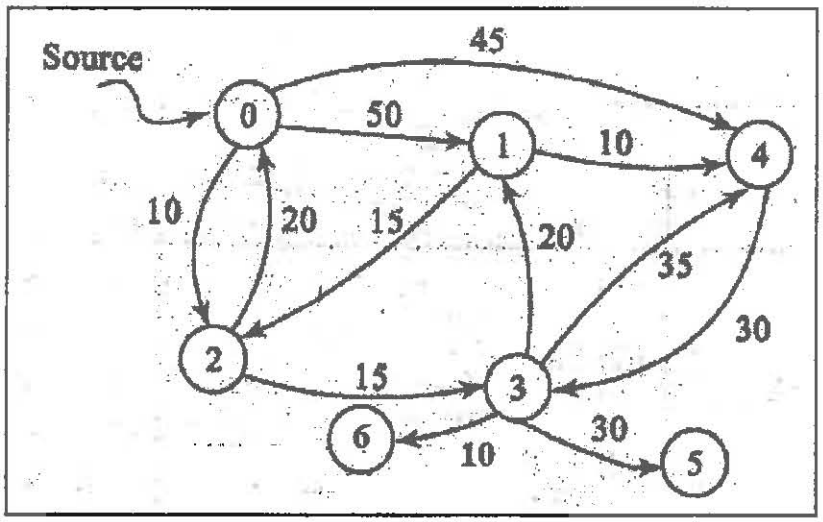

Figure 4.3
[Rajah 4.3]

[8 Marks/*Markah*]

Hints Answer Table:

Iteration	T	u	DISTANCE				
			[1]	[2]	[3]	[4]	[5]

The shortest Path and Distance

Source	Destination	Shortest Path	Shortest Distance

QUESTION 5
[SOALAN 5]

(C3, CO3, PO3, PO4, PO11)

(a) Consider a list L={22, 54, 65, 76, 77, 87, 88, 93, 97, 99} of ordered numerical elements. Demonstrate the Interpolation Search technique for the elements 54 and 94.

[Pertimbangkan senarai L = {22, 54, 65, 76, 77, 87, 88, 93, 97,99} yang merupakan unsur tertiban berangka. Tunjukkan teknik Carian Interpolasi untuk unsur-unsur 54 dan 94.]

[6 Marks/*Markah*]

(C3, CO3, PO3, PO4, PO11)

(b) Consider a list L = {69, 15, 85, 89, 12, 26}. Demonstrate and compare the Selection Sort and Quick Sort technique on the list L.

[Pertimbangkan senarai L = {69, 15, 85, 89,12, 26}. Tunjukkan dan bandingkan Isihan Pemilihan dan teknik Isihan Pantas senarai L.]

[14 Marks/*Markah*]

--- oooooOooooo---

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

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