



UNIVERSITI TUN HUSSEIN ONN MALAYSIA

**PEPERIKSAAN AKHIR
SEMESTER II
SESI 2009/2010**

SUBJECT NAME : DATA STRUCTURE & ALGORITHM
SUBJECT CODE : BIT 1073
COURSE : 1 BIT
DATE OF EXAMINATION : APRIL/MAY 2010
INSTRUCTION : 2 HOURS 30 MINUTES
: ANSWER ALL QUESTIONS. IN
SECTION A AND B AND **FOUR (4)**
QUESTIONS IN SECTION C

THIS QUESTION PAPER CONTAINS TWELVE (12) PAGES

SECTION A

Instruction: Answer **ALL** questions.

- Q1** Suppose cursor points to a node in a list (using the IntNode class with instance variables called data and link). What statement changes cursor so that it refers to the next node?

The IntNode definition:

```
public class IntNode
{
    int data;
    IntNode link;
}
```

- (a) cursor++;
 - (b) cursor = link;
 - (c) cursor += link;
 - (d) cursor=cursor -> link;
- Q2** Suppose cursor refers to a node in a linked list (using the Node class with instance variables called data and link). What Boolean statement will be true when cursor refers to the tail node of the list?
- (a) (cursor == null)
 - (b) (cursor->link == null)
 - (c) (cursor->data == null)
 - (d) (cursor->data == 0.0)
- Q3** The operation for adding an entry to a stack is traditionally called
- (a) add
 - (b) append
 - (c) insert
 - (d) push

Q4 In the linked list implementation of the stack class, where does the push member function place the new entry on the linked list?

- (a) at the head
- (b) at the tail
- (c) after all other entries that are greater than the new entry
- (d) after all other entries that are smaller than the new entry

Q5 Suppose we are sorting an array of eight integers using some quadratic sorting algorithm. After four iterations of the algorithm's main loop, the array elements are ordered as below:

2 4 5 7 8 1 3 6

- (a) The algorithm might be either selection sort or insertion sort
- (b) The algorithm might be selection sort, but it is not insertion sort
- (c) The algorithm is not selection sort, but it might be insertion sort
- (d) The algorithm is neither selection sort nor insertion sort

Q6 Link to linked list in C is of type:

- (a) unsigned integer
- (b) pointer to integer
- (c) pointer to struct
- (d) none of the above

Q7 What is the worse-case for serial search finding a single item in an array?

- (a) constant time
- (b) logarithmic time
- (c) linear time
- (d) quadratic time

Q8 Consider the following statements:

```
int *p;  
int i;  
int k;  
i = 42;  
k = i;  
p = &i;
```

After these statements, which of the following will change the value of *i* to 75?

- (a) `k = 75;`
- (b) `*k = 75;`
- (c) `p = 75;`
- (d) `*p = 75;`

Q9 Consider the following statements:

```
int i = 42;  
int j = 80;  
int *p1;  
int *p2;  
p1 = &i;  
p2=&j;  
*p1 = *p2;  
printf( " %d %d", i, j);
```

What numbers are printed as the output statements?

- (a) 42 and then another 42
- (b) 42 and then 80
- (c) 80 and then 42
- (d) 80 and then 80

Q10 A diagraph in which outdegree is the same as indegree is called

- (a) balanced
- (b) symmetric
- (c) regular
- (d) none of these

SECTION B

Instruction: Determine whether the following statements are **TRUE** or **FALSE**.

- Q11** Stack is an example of linear list.
- Q12** Pop operation in stack may result in overflow.
- Q13** A queue can be implemented using a circular array with front and rear indices and one position left vacant.
- Q14** A binary search of an ordered set of elements in an array is always faster than a sequential search of the element.
- Q15** A complete graph is one in which every node is connected to every other node.
- Q16** A tree node can have only two children.
- Q17** A recursive function must return a value to its calling function; otherwise, it cannot continue the recursive process.
- Q18** Diagonal entries in adjacency matrix for directed graph can be non-zero entries.
- Q19** The adjacency matrix of a directed graph need not be symmetric.
- Q20** Depth-first search (DFS) has the same complexity as breadth-first search (BFS).

SECTION C

Instruction: Answer **FOUR (4)** questions only.

- Q21** (a) Suppose the following list of letters is inserted in order into an empty binary tree:

J R D G T E M H P A F Q

- (i) Draw the final tree. (4 marks)
- (ii) The value of **INORDER**, **PREORDER** and **POSTORDER** traversal. (6 marks)
- (b) Convert the expression $((A+B) * C - (D-E) ^ (F+G))$ to equivalent :
- (i) Prefix notations (5 marks)
- (ii) Postfix notations. (5 marks)

- Q22** (a) Consider the following snippet of C code:

```
for (int x=0; x<n; x++)
{
    int min = x;
    for (int y=x; y<n; y++)
    {
        if (array[y] < array[min])
            min = y;
    }
    int temp = array[x];
    array[x] = array[min];
    array[min] = temp;
}
```

- (i) What is the complexity of this algorithm? (3 marks)
- (ii) What is this search called? (2 marks)

(b) Consider the following two Java functions.

```
public static void stars(int N) {
    for (int i = 0; i < N; i++)
        System.out.print("*");
    System.out.println();
}

public static void mystery(int N) {
    if (N == 0) return;
    stars(N);
    mystery(N-1);
    stars(N);
}
```

(i) What gets printed when `mystery(3)` is called?

(5 marks)

(ii) What gets printed when `mystery(4)` is called?

(5 marks)

(c) Consider the following snippet of C code:

```
a(int i)
{
    int j;
    j = i*5;
    printf("In procedure a: i=%d, j=%d\n",i,j);
    if (i > 0) a(i-1);
    printf("Later In procedure a:i=%d, j=%d\n",i,j);
}

void main()
{
    int i;
    i = 16;
    a(3);
    printf("main: %d\n",i);
}
```

What is the printed output?

(5 marks)

Q23 Consider the following directed graph in **FIGURE Q23(a)**:

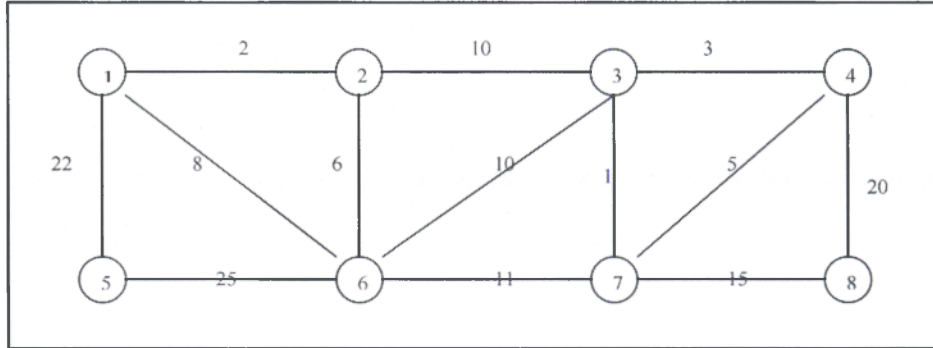


FIGURE Q23(a)

- (a) A minimum-cost spanning tree is to be constructed using Kruskal's method.
- (i) Describe the working of Kruskal's method on this graph. (3 marks)
 - (ii) Draw the spanning tree and explain each step. (4 marks)
 - (iii) What is the cost of the constructed tree? (3 marks)

(b) Based on **FIGURE Q23(b)**, traverse the graph using Breadth First Traversal starting from Frankfurt by showing:

- (i) The visited cities.
- (ii) Content of your list (please mention if you are using either stack or queue).

(10 marks)

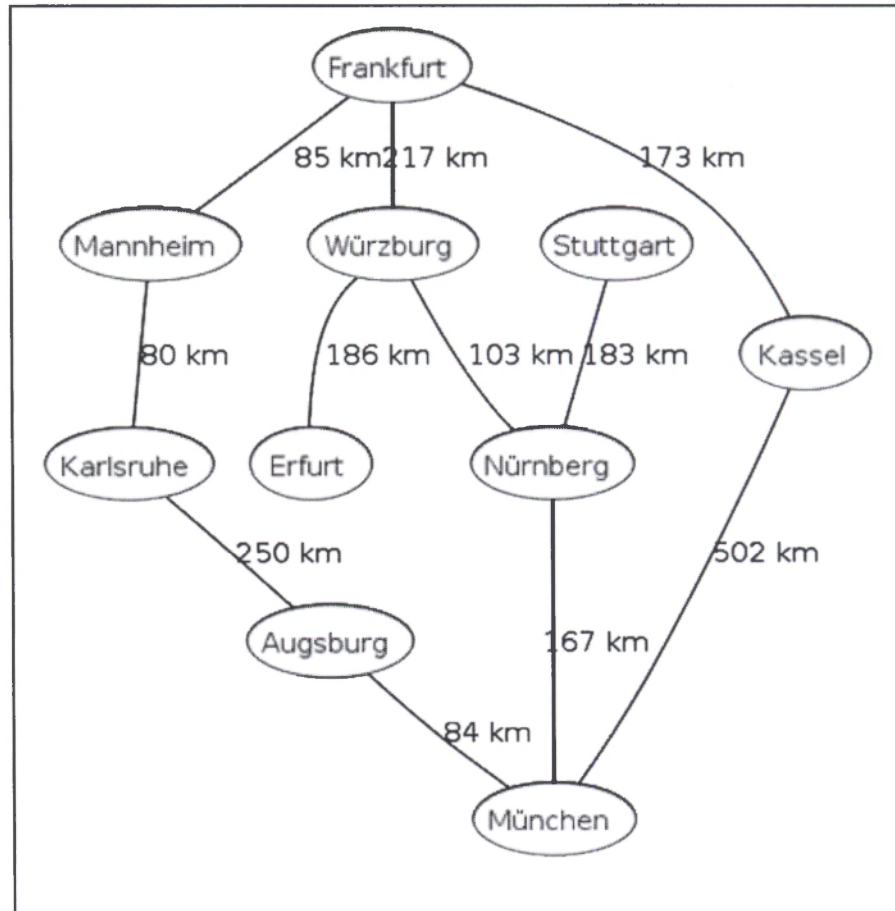


FIGURE Q23(b)

Q24 (a) By referring to **FIGURE Q24**, answer the questions below:

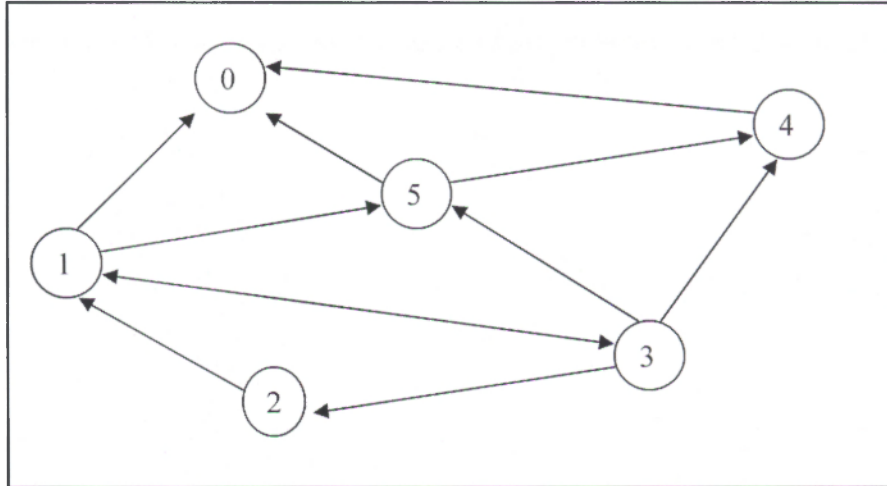


FIGURE Q24

- (i) Indegree and outdegree of each vertex. (3 marks)
- (ii) Adjacency matrix. (3 marks)
- (iii) Adjacency list representation. (3 marks)
- (iv) How many edges does an n -vertex undirected graph needs in order to make it a complete graph? (3 marks)
- (b)** Which sorting algorithm does these distinct descriptions fits into?
- (i) choose a pivot value
 partition the array:
 left part has items \leq pivot
 right part has items \geq pivot
 recursively sort the left part
 recursively sort the right part
 Algorithm Complexity: worst-case $O(N^2)$, expected $O(N \log N)$ (2 marks)

- (ii) N passes
on pass k: find the kth smallest item, put it in its final place
Algorithm Complexity: always $O(N^2)$
(2 marks)
- (iii) N passes
on pass k: insert the kth item into its proper position relative to the items to its left
Algorithm Complexity: worst-case $O(N^2)$, given an already-sorted array: $O(N)$
(2 marks)
- (iv) recursively sort the first $N/2$ items
recursively sort the last $N/2$ items
Algorithm Complexity: always $O(N \log N)$
(2 marks)

Q25 (a) By using `struct`, write a program, which will accept name, ID number and test marks from Test 1 to 5. Your program should be able to print the average marks for each students. The sample of the output is shown below:

Output sample:

```

Student Name      : AINA BINTI HUSIN
ID No.           : M8989
Test 1           : 78
Test 2           : 98
Test 3           : 67
Test 4           : 84
Test 5           : 90
Average          : 83.4
    
```

} INPUT
(by keyboard)

} OUTPUT

(10 marks)

(b) Write a program to create a binary tree with a single node containing `int` data 55 and two (2) pointers. The first node address should be called as `root`.
(6 marks)

(c) By referring to **FIGURE Q25(b)**, show what would happen if the following statements are executed by showing your answer in graphically.

1. `temp = plist`
2. `loop (temp -> link not null)`
 - a. `temp = temp -> link`
3. `temp->link = plist`

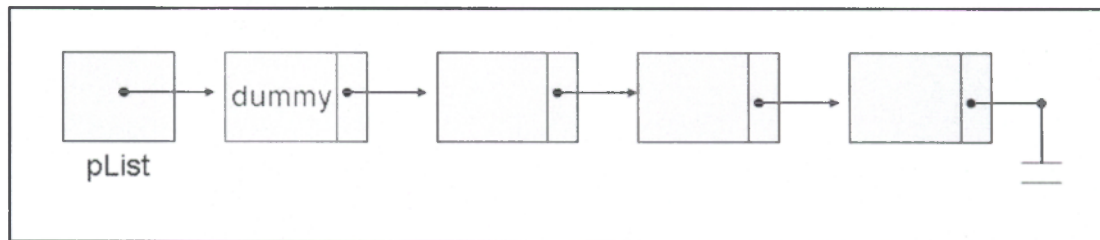


FIGURE Q25(b)