National School of Business Management

Algorithms and Data Structures
CS106.3 - 16.2 Examination

Time: 03Hrs

Answer all Questions **Date: 2017**

Question 1 – 20 Marks

(a) What is asymptotic analysis in the context of evaluation of Algorithms? [5 Marks]

(b) Simplify the following Big-O expressions [10 Marks]

i. $O(10n^4-n^2-10)$

ii. $O(2n^3+10^2n) * O(n)$

iii. $O(n^2) + O(5*\log(n))$

iv. $O(n^2) * O(5*\log(n))$

v. $n^2*O(n^2)$

(c) Giving reasons, evaluate the time complexity of the following function. [5 Marks]

```
int swapping;
do {
   for(i=0,swapping=0;i<size-1;i++)
        if(data[i]>data[i+1]) {  temp=data[i];
                                 data[i]=data[i+1];
                                 data[i+1]=temp;
                                 swapping=1;
                              }
} while(swapping)
```

Question 2 - 20 Marks

Following loop implements the insertion sort algorithm.

```
for(k=1; k<size; k++){
     for(i=0; i<k && d[i]<=d[k]; i++);
     temp=d[k];
     for(l=k;l>i;l--) d[l]=d[l-1];
     d[i]=temp;
}
```

(a) Write a complete C function to implement the insertion sort algorithm. [5 marks]

(b) Write down a comment line for each and every line in the above C function in part (a). If necessary number the lines in the C function and write the comments separately with the line numbers. [5marks]

(c) Copy the following table into your answer script and complete it for each iteration of the outer loop (loop using k) for the problem scenario given below to carry out a desk-check of the code given above.

Variable	initially	After iteration 1	After iteration 2	After iteration 3	After iteration 4	…
`Array d[]`	12,25,9,2, 20,15,8					
`size`	7					
`k`	1					
`d[k]`	25					
`i`	NA					
`d[i]`	NA					

Problem Scenario:

`array[]`

12	25	9	2	20	15	8

[10 Marks]

Question 3 - 20 Marks

(a) Show diagrammatically, the operation of a queue implemented via a circular array assuming the array size to be 8. Draw diagrams to show the status of the queue for the following, clearly indicating the position of the head/front and tail/rear.

i. empty queue

ii. after enqueueing 4 items; 12, 23, 8 and 15

iii. full queue [6 Marks]

(b) Suggest a method to differentiate an empty queue (in section (a) above) from a full queue by comparing the values of head and tail. [5 Marks]

(c) Implement 3 functions, including either enqueue() or dequeue(), of the queue specification given below. Utilize your suggestion in part (b) above. [9 Marks]

```
     struct queue {      int head, tail;
                         int data[size];     };
     int enqueue(struct  queue *q, int item);
     int dequeue(struct queue*q);
     int init(struct queue* q);    // set head and tail
     int full(struct queue* q);    // return 1 if full
     int empty(struct queue* q);  // return 1 if empty
```

Question 4 - 20 Marks

Following incomplete code segment intends to implement the linear search algorithm.

```
int lsearch(float data[], int size, float key){
     int found = 0;
     int position = -1;
     int index = 0;
```

(a) Complete the above code to using appropriate syntax. [6 marks]

(b) Write down a comment line you would include in the above code against each line to illustrate the function of each line or statement. You do not have to copy the code – just put the line number and your comment in your answer script. [4 marks]

Following lines were extracted from a typical linked list implementation prototype in C.

```
// data structure for a node
struct node {  float data;
               struct node* next;};

//creates a node with data value (item) & returns the node address
struct node*  makeNode(float item);

// add a new node to the head
struct node*  addHead(struct node** list, float item);

// add a new node to the tail
struct node*  addTail(struct node** list, float item);
```

(c) Write code for the above three functions to derive the linked list implementation. [10 marks]

Question 5 - 20 Marks

(a) Draw a binary search tree (BST) generated by inserting the following items in the given order.
54, 15, 12, 64, 51, 9, 85, 15, 24, 3 [6 Marks]

(b) Draw the sequence of items you process, if the BST is traversed by,
i. pre-order,
ii. in-order,
iii. post-order, tree walking methods. [6 marks]

(c) Write down a node structure in C, suitable to implement the above BST. [2 marks]

(d) Write a C function to return the minimum value stored in a BST. [4 Marks]

(e) Explain the use of tree data structure in a computer application. [2 Marks]