187

B.Sc. Engg. CSE 3rd Semester **12 March 2019 (Morning)**

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

MID SEMESTER EXAMINATION **WINTER SEMESTER, 2018-2019**

DURATION: 1 Hour 30 Minutes **FULL MARKS: 75**

CSE 4303: Data Structures

Programmable calculators are not allowed. Do not write anything on the question paper.

There are **4 (four)** questions. Answer any **3 (three)** of them.

Figures in the right margin indicate marks.

---

1. a) Consider the *List* class with the following *private* members: 10

```
class List{
private:
      struct Node{
      int item;               // the data of the node
      Node* next;             // points to the next node of the list
      };
Node* head;                  // point to first node in the list
};
```

Consider the linked list of *integers* represented by the following diagram:

$$\begin{tikzcd}
\overset{\text{head}}{\boxed{\bullet}} \arrow[r] & \boxed{5 \mid \bullet} \arrow[r] & \boxed{3 \mid \bullet} \arrow[r] & \boxed{7 \mid \bullet} \arrow[r] & \boxed{12 \mid \bullet} \arrow[r] & \boxed{10 \mid \text{NULL}}
\end{tikzcd}$$

i. Draw a diagram of the above list after the following lines of code have been executed:

```
Node* prev = head->next;
Node* nodeToInsert = new Node;
nodeToInsert->item = 4;
nodeToInsert->next = prev->next;
prev->next = nodeToInsert;
```

ii. Assume that the code represented above in part (i) has executed. What is the value of *prev->item*?

iii. In addition to the code above, assume the following code executes. Draw a diagram of the list after this code executes as well.

```
prev = prev->next;
prev = prev->next;
Node* curr = prev->next;
prev->next = curr->next;
delete curr;
curr = NULL;
```

b) Bob writes down a number between 1 and 1,000. Mary must identify that number by asking "yes/no" questions to Bob. Mary knows that Bob always tells the truth. If Mary uses an optimal strategy, then she will determine the answer at the end of exactly how many questions in the worst case. 7

c) Discuss different types of memory allocation techniques with appropriate example. 8

2. a) Define Complexity. Translate Q into its equivalent postfix expression P using stack. 9

$$Q = A * (B + D) / E - F - (G + H / K)$$

b) Assume you have a stack with operations: *push(), pop(), top(), isEmpty()*. How would you use these stack operations to simulate a queue, in particular, the operations: *enqueue()* and *dequeue()*? 8

c) On each iteration of its outer loop, insertion sort finds the correct place to insert the next item, relative to the ones that are already in sorted order. It does this by searching back through those items, one at a time. 4+4

   i. Would insertion sort be speeded up by using binary search to find the correct place to insert the next item? Justify your answer.

   ii. What is the running time for insertion sort when the array is already sorted in ascending order and descending order?

3. a) Analyze the complexity of bucket sort and radix sort. Briefly mention the limitations of these two sorting algorithms. 8

b) What is the best asymptotic ("big-O") characterization of the following functions: 10

   i. $f(n) = 2^5 + 5n^3\log(n) + 2^6n^2 + 100\,n^4$

   ii. $f(n) = 1000n^2 + 16n + 2^n$

   iii. $f(n) = n + (n-1) + (n-2) + \cdots + 3 + 2 + 1$

   iv. $f(n) = 2^{10} + 3^5$

   v. $f(n) = 37n + n\log(n^2) + 5000\log(n)$

c) Consider the following numbers to construct a binary heap tree. 3+4

18 , 16 , 14 , 12 , 10 , 8 , 6 , 4 , 2

   i. After inserting all the numbers sequentially, insert a new item *15*. Now perform *heapify* (if necessary). Show step by step.

   ii. Use the binary heap tree you created in (i). Now delete the item *18* and balance the tree by following the conditions of a binary heap tree.

4. a) Consider the following list of numbers: 4+4

62 , 31 , 70 , 91 , 25 , 11 , 9 , 61 , 73 , 6

   i. Show the resulting tree after inserting the numbers in the same order specified above into an initially empty binary search tree.

   ii. Use the binary search tree you created in (i). What are the two possible binary search trees after 62 is deleted.

b) Discuss three different ways of improving the performance of the basic Bubble sort algorithm with suitable example. 9

c) Explain Time-Space Tradeoff with a suitable example. Analyze the code of Fig. 2 and find out the complexity in terms of Big-O notation step by step. (try to make the upper bound tighter) 8

```
void complexity(){
     int n, val;
     for ( int j = 4; j < n; j = j+2 ){
          val = 0;
          for ( int i = 0; i < j; ++i){
               val = val + i * j;
               for ( int k = 0; k < n; ++k){
                    val++;
               }
          }
     }
}
```

Figure 2: Figure for the question no. 4 (c)