00439

Sri Lanka Institute of Information Technology

B.Sc. Honours Degree in Information Technology

Specialized in Information Technology

Final Examination (June Intake)
Year 2, Semester 2 (2023)

IT2070 – Data Structures and Algorithms

Duration: 2 Hours

June 2023

Instructions to Candidates:

- This paper has 4 questions.
- Answer all questions in the booklet given.
- The total marks for the paper is 100.
- This paper contains 9 pages, including the cover page.
- Electronic devices capable of storing and retrieving text, including calculators and mobile phones are not allowed.
- 10 minutes reading time is allowed

Question 1 **(25 marks)**

a) What are the basic features in the following data structures in relation to the basic operations insert, delete and retrieval data. (3 marks)

	Insert	Delete	Retrieve
Stack			
Linear Queue			
Circular Queue			

b) Fill in the following blanks. (6 marks)

i) Most popular data structure is _____.

ii) The process of retrieving the element at the top of a stack without removing it is called_____

iii) A stack can be implemented using _____ or _____

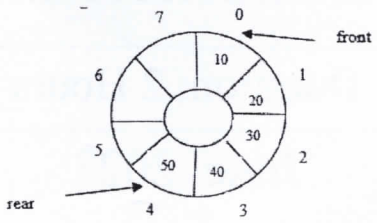
iv) In a stack, if the "top" pointer is equal -1, then the stack is _____

v) The time complexity of the push and pop operations on a stack implementation are; push -_____ and pop - _____

c) Consider the following circular queue with initial values given below.

Initials values are :

Front=0
Rear=4
Count=5



Show how the above parameter values (Front, Rear and Count) will change after each of the operations by completing the table given below. (6 marks)

i) insert(60)
ii) insert(70)
iii) delete( )
iv) insert(80)
v) insert(90)
vi) insert(100)

Operation	**Front**	**Rear**	**Count**
insert(60)			
insert(70)			
----------			

d) Consider the UML class diagrams given below.

StackX
- double[] stackArray - int maxSize - int top
+void push(double j) +double pop( ) +double peek( ) +boolean isEmpty( ) +boolean isFull( )

QueueX
- StackX s1 - StackX s2
+ void insert(double j) + double remove( ) + double peekFront( ) + boolean isEmplty( ) + boolean isFull ( )

A queue data structure can be implemented using two stacks. Assume the size of these stacks as 5. Assume that the java implementation of the StackX class is given.

Complete the following java code for the QueueX class. (6 marks)

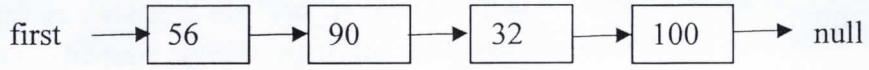
```
public class QueueX
{
     private StackX s1;
     private StackX s2;

     public QueueX(int x)
     {
          ..........//Line 1
          ..........//Line 2
     }

     public void insert(double j)
     {
          ..........//Line 3 - insert an element to stack 1
     }

     public double remove( )
     {
          if( s2.isEmpty( )){
               while(!s1.isEmpty()){
                    s2.push( s1.pop())
               }

          }
          ..........//Line 4
     }
```

e) Consider the Queue implementation given in part d) above, draw the status of the two stacks, after the execution of the following code. (4 marks)

```
class QueueApp {
     public static void main(String[] args) {
          QueueX q = new QueueX(5);
          q.insert(5.0);
          q.insert(3.0);
          double temp = q.remove( );
          q.insert(10.0);
     }
}
```

Question 2 **(25 marks)**

a) Consider the below link list (4 marks)



Display the output and draw the link list after executing the following code segments

```
first = first.next;
System.out.print(first.next.ID);
first.next = NULL;
```

b) Consider the below Link class and the LinkList class. Assume the classes have already been implemented. addFirst() method in the LinkList class inserts a new link as the first link and removeFirst() method deletes the first link of the link list.

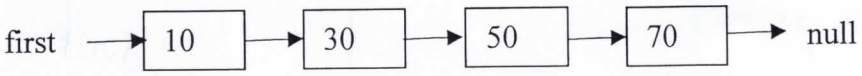
Link
+int ID +Link next
+ Link (int IDNo)

LinkList
-Link first
+void LinkList() +boolean isEmpty() +void addFirst(int ID) +int removeFirst()

i) Modify the LinkList class by adding another method called deleteAllLinks(). This method deletes all the links in the link list and display the IDs of the deleted links. Implement the method deleteAllLinks(). (5 marks)

ii)

1) Consider the below link list. What is the output you get after calling deleteAllLinks() method? (2 marks)



2) Draw the above link list after executing deleteAllLinks() method. (2 marks)

iii) Modify the method implemented in b) i) to remove all the links except the last link. (3 marks)

c) Key values of a binary search tree is given below.

90, 34, 100, 23, 50, 98, 125, 10

i) Insert the above values in a binary search tree. (2 marks)
ii) What type of a binary tree did you get in c) i)? (1 mark)
iii) How do you convert the above tree to a full binary tree? (1 mark)
iv) Analyze and discuss the importance of a full binary tree when searching a value from a tree data structure (5 marks)

Question 3 **(25 marks)**

a) Consider the following pseudocode and find the complexity using Big O notation. (3 marks)

```
for (i = 1 to 5)
     print i
for (j = 1 to n)
     print j
```

b) PARTITION() algorithm which is used in Quick Sort algorithm is given below. It returns the partition index.

i) Consider the following array A and illustrate the steps of applying PARTITION() algorithm to array A (6 marks)

Array A

1	2	3	4	5
5	4	0	8	6

```
PARTITION(A, p, r)
1   x = A[r]
2   i = p - 1
3   for j = p to r - 1
4     if A[j] ≤ x
5           i = i + 1
6           exchange A[i] with A[j]
7   exchange A[i + 1] with A[r]
8   return i + 1
```

ii) Which value has been selected as the pivot element? (2 marks)

iii) What is the partition index computed by the partition algorithm? (3 marks)

c)

i) What is the best case in Quick Sort algorithm? (2 marks)

ii) Obtain the recurrence equation for the best case in Quick Sort algorithm? (2 marks)

iii) Master Theorem can be applied to recurrences of the form $T(n) = a\,T(n/b) + f(n)$. Solve the recurrence equation obtained in c) ii) using Master Theorem and find the complexity. (3 marks)

Master Theorem

$$T(n) = \begin{cases} \Theta\left(n^{\log_b a}\right) & f(n) = O\left(n^{\log_b a - \varepsilon}\right) \rightarrow f(n) < n^{\log_b a} \\ \Theta\left(n^{\log_b a} \lg n\right) & f(n) = \Theta\left(n^{\log_b a}\right) \rightarrow f(n) = n^{\log_b a} \\ \Theta\left(f(n)\right) & f(n) = \Omega\left(n^{\log_b a + \varepsilon}\right) \rightarrow f(n) > n^{\log_b a} \\ & \text{if } af(n/b) \le cf(n) \text{ for c} < 1 \text{ and large } n \end{cases}$$

```
QUICKSORT (A,p,r)
 1 if p < r
 2    q = PARTITION(A, p, r)
 3    QUICKSORT (A, p, q-1)
 4    QUICKSORT (A, q+1, r)
```

d) Figure (1) is the pseudocode of the Insertion Sort algorithm. Describe the impact of the output and the performance if line (7) of the same pseudocode is changed as shown in Figure (2). (Line 7 in Figure(2) is inside the while loop) (4 marks)

```
INSERTION-SORT(A)
1 for j = 2 to A.length
2     key = A[j]
3     i = j - 1
4     While i > 0 and A[i] > key
5       A[i+1] = A[i]
6       i = i-1
7     A[i+1]  = key
```

Figure (1)

```
INSERTION-SORT(A)
1 for j = 2 to A.length
2     key = A[j]
3     i = j - 1
4     While i > 0 and A[i] > key
5       A[i+1] = A[i]
6       i = i-1
7       A[i+1]  = key
```

Figure (2)

Question 4 **(25 marks)**

a) What is a max heap? Explain your answer using an example. (3 marks)

b) Identify the violating node of the below max heap and illustrate the operations of the MAX_HEAPIFY (A,i) for the array A given below. (Use diagrammatic way to reach the answer) (3 marks)

Array A

60	40	58	10	44	55	33	2	6	42	15

Pseudocode of MAX_HEAPIFY is given below

```
MAX_HEAPIFY (A,i)
1.    l =  LEFT_CHILD (i);
2.    r =  RIGHT_CHILD (i);
3.    if l ≤ heap_size[A] and A[ l ] > A[ i ]
4.          largest = l;
5.    else largest = i;
6.    if r ≤ heap_size[A] and A[r] > A[largest]
7.          largest =  r;
8.    if largest ≠  i
9.          exchange A[i] ↔ A[largest]
10.         MAX_HEAPIFY (A, largest)
```

c) Consider the HEAPSORT algorithm is given below.

```
HEAPSORT(A)
1.    BUILD_MAX_HEAP[A]
2.    for  k = A.length down to 2
3.         exchange A[1] with A[k]
4.         A.heap_size = A.heap_size - 1;
5.         MAX_HEAPIFY(....., .....)
```

i) Complete line number 5 of the HEAPSORT algorithm. (2 marks)

ii) Explain the reason for the calling **MAX_HEAPIFY** algorithm in line number 5 with the arguments given in c) i) above. (4 marks)

iii) A student has written the above heapsort algorithm by changing line number 2 as follows,

```
for  k = A.length down to 1
```

Discuss how this modification affects the results and the performance (5 marks)

d)

i) Draw the state transition diagram for a string-matching automation for the pattern P = abba and take the input alphabet as {a,b} (6 marks)

ii) Consider the below text

a	b	b	a	b	b	a	a	b	a

Illustrate how you would find the pattern "abba" in the above text using your answer in part d) i) above. (2 marks)

End of Question Paper