00035

Sri Lanka Institute of Information Technology

B.Sc. Honours Degree in Information Technology

Specialized in Information Technology

Final Examination
Year 2, Semester 2 (2019)

IT2070 – Data Structures and Algorithms

Duration: 2 Hours

October 2019

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.
- This paper is preceded by 10 minutes reading period. The supervisor will indicate when answering may commence.

Question 1 (25 marks)

a) A Stack class called **StackX** has been created to store 5 integer values. push(), pop(), isEmpty(), isFull() and size() methods are available in the class.

i) Draw separate stack frames after performing the below operations to an empty stack. (4 marks)

```
s.push(5);
s.push(s.peek());
s.push(s.pop());
s.pop();
```

ii) Find the errors in the push method implemented below and correct them (3 marks)

```
public void  push(int j) {
        if (top == maxSize)
                system.out.println("Stack is full");
        else {
                stackArray[top] = j;
                top++;
        }
}
```

iii) Numbers are stored in a stack object called **s1**. Write the code segment written in main application to remove the odd numbers from the stack. Even numbers should remain in the stack. (5 marks)

b)

i) Draw the queue frames after performing the given code segment. A circular queue is used. (4 marks)

```
QueueX Q = new Queue(10);
int no = 10;
for(int i=1; i<=8; i++) {

   Q.insert(no);
   no = no + 10;
}
for(int i=1; i<=4; i++) {

     Q.insert(Q.remove());
}
```

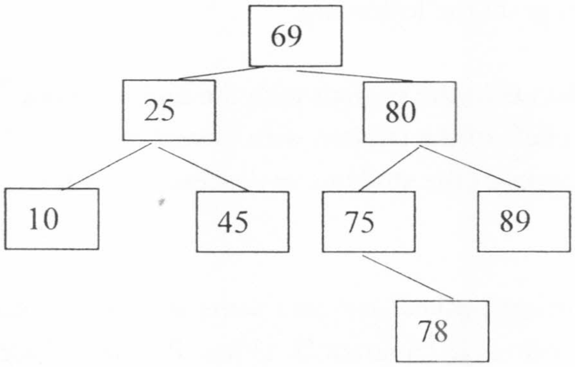
ii) What is the result you get if you remove all the items from the above queue? (2 marks)

iii) getSize() method of a circular queue class returns the no of items in the queue. Implement getSize() method of the circular queue class. (2 marks)

c) Assume a queue class and stack class have been implemented. Using them, write code segment to reverse the given items in a queue. (i.e. After executing the code the items in the queue should be reversed)
Assume myStack and myQueue objects have already been created using the implemented classes and values are already available in the queue.
(5 marks)

Question 2 **(25 marks)**

a) Consider the following binary search tree and answer the questions



i) If the value 72 is inserted into this tree, which node becomes its parent? (1 mark)
ii) If the value 48 is inserted into this tree, which node becomes its parent? (1 mark)
iii) List all non leaf nodes in this tree. (1 mark)
iv) If node 25 is deleted which node should be its replacement node? (1 mark)
v) Find the depth of this tree (1 mark)
vi) Is this a complete binary tree (1 mark)

b) Consider the following classes to answer the below question. Assume the following classes have been implemented

Student class

int studentID
int marks
Student leftChild;
Student rightChild;

void displayDetails();

Tree class

Student root
void insert(int marks, int studentID) Student minimum( ) Student maximum ( ) void descendingOrder ( )

Students of a class are stored in a tree data structure. Mark of the student is considered as the key to store them in the tree. Assume the tree object **studentMarks** is created with student details.

Write a code segment to do the following,

i) Display the details of the student with the highest mark (2 marks)

ii) Display the details of the student with lowest mark (2 marks)

iii) Display the marks of the students in descending order. (2 marks)

c) i) Consider the following LinkedLists and write a code segment that would change the LinkedList from Diagram A to Diagram B. (First, P1, and P2 are references to given links) (3 marks)

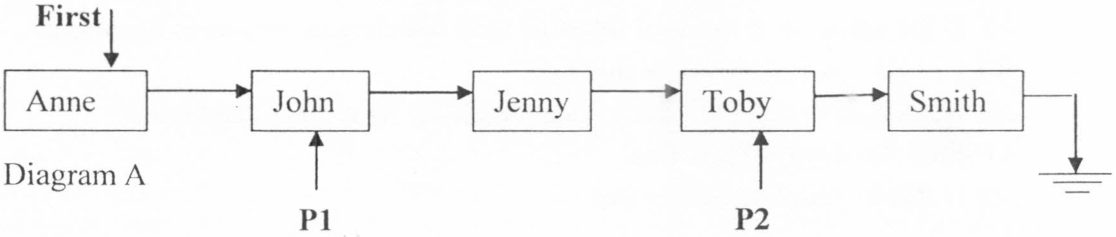


Diagram A

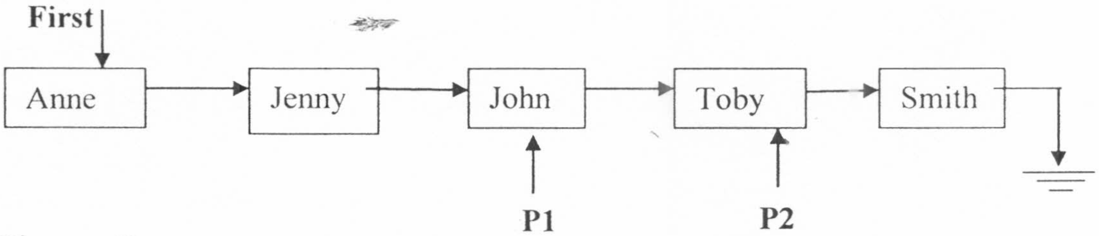
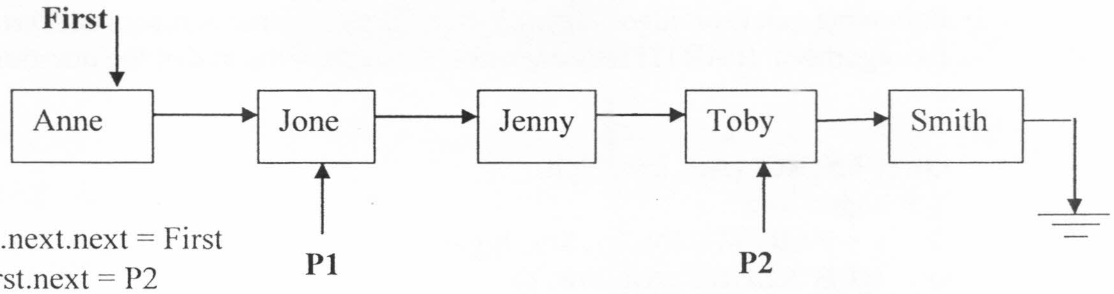


Diagram B

ii) Consider the following linked list. Illustrate how the linked list changes when each code segment is executed on the linked list. (3 marks)



P1.next.next = First
First.next = P2
First = P2.next
First.next = P1
P2.next = NULL

d) A library maintains the details of their books using a linked list. A book has a

Book No	integer
Number of copies	double

Assume the **book** class and the **linked list** class has already been implemented.

i) Implement a method called isAvailable() in the linked list class to return the reference of the link if the library has a given book. The book number is given as a parameter. (4 marks)

ii) Implement a method called lending( ) in the linked list class to update the number of copies (decrement by 1) when the book number is given as a parameter.
Hint : use the isAvailable( ) method to check the availability. (3 marks)

Question 3 (25 marks)

a) What is the running time of the below method using Big O notation? Explain your answer. (4 marks)

```
public int add100(int[] array) {

    if (array.length < 100)
       return 0;

   int sum = 0;
   for (int i = 0; i < 100; i++)
      sum += array[i];

   return sum;
}
```

b)

i) Following quicksort algorithm has errors. State the line numbers with errors and rewrite the algorithm. (PARTITION algorithm is given at the end of the question paper) (4 marks)

```
QUICKSORT (Arr, low, high)
1 if high < low
2     p = PARTITION(Arr, low, high)
3     QUICKSORT (Arr, low, p)
4     QUICKSORT (Arr, p, high)
```

ii) Describe the worst case of the of the quick sort algorithm using a diagram and find its recurrence equation. (2 marks)

iii) Solve the above equation using repeated substituted method and find the worst case running time in Theta notation. (4 marks)

c)

i) What is a max heap property? (1 mark)

ii) Consider the following array representation of a heap

A

60	50	45	30	20	35	15	10

Is this a max heap? Justify your answer. (3 marks)

d) Consider the below HEAP_EXTRACT_MAX and MAX_HEAPIFY algorithms used to return the maximum value from a max-heap.

Convert them to **HEAP_EXTRACT_MIN** algorithm which would remove and return the minimum value from a min heap. (7 marks)

```
HEAP_EXTRACT_MAX(A[1 .. n])
if A.heap_size >= 1
    max = A[1]
    A[1] = A[A.heap_size]
    A.heap_size = A.heap_size -1
    MAX_HEAPIFY(A,1)
    return max
```

```
MAX_HEAPIFY (A,i)
l = LEFT(i);
r = RIGHT(i);
if l ≤ A.heap_size and A[ l ] > A[ i ]
        largest = l;
else largest = i;
if r ≤ A.heap_size and A[r] > A[largest]
        largest = r;
if largest ≠ i
        exchange A[i] with A[largest]
        MAX_HEAPIFY (A,largest)
```

Question 4 **(25 marks)**

a) Following is the **Naïve-String-Matcher** algorithm, which is used to find the occurrence(s) of a pattern string within another string or body of text.

```
Naïve-String-Matcher (T, P)
    n = T.length
    m = P.length
        for s = 0 to n-m
            if P[1..m] = T[s+1..s+m]
            then print "Pattern occurs with shift" s
```

Given the text and pattern as follows;

Text ***T***

a	b	b	a	c	d	a	a	b

Pattern ***P***

a	a	a

i) How many comparisons would occur in this algorithm? (2 mark)

ii) How many valid and invalid shifts would occur in this algorithm? (2 marks)

b)

i. If modulo value is $q = 100$, how many spurious hits and valid hits do the **Rabin -Karp matcher** encounter in the text T = 203410052006 when looking for pattern $P = 100$? (4 marks)

ii. How do you reduce the number of spurious hits in b) i)? (2 marks)

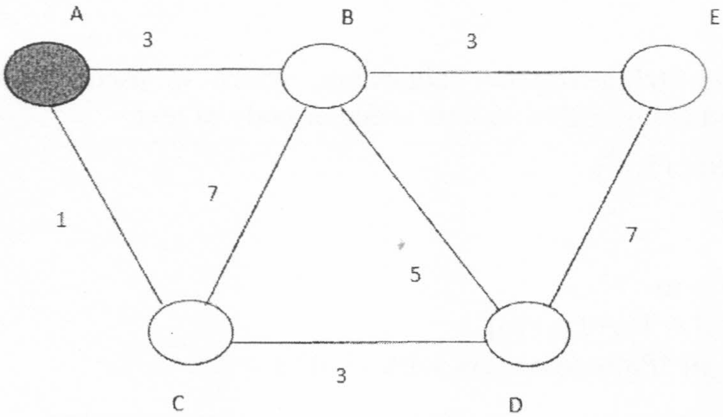
iii. What should be situation of the worst-case scenario occurs in Rabin-Karp algorithm? (2 marks)

c) Draw the state transition diagram for a string-matching automation for the pattern P = ***aba*** and take the input alphabet as $\{a, b, c\}$

(7 marks)

d) What is meant by the "**Optimal Solution**" in Greedy method? (1 mark)

e) Apply the **Dijkstra's** algorithm given at the end of the paper to find the shortest path from the source vertex ***s*** to all the other vertices of the graph. (For the purpose of illustration, assign the values only once to the given algorithm and use diagrammatic way to reach the answer.) (5 marks)



```
DIJKSTRA($G, w, s$)
  for each vertex $v \in V[G]$
      $d[v] = \infty$
      $\pi[v]$ = NIL
  $d[s] = 0$
  $S = \emptyset$
  $Q = V[G]$
  while $Q \neq \emptyset$
      $u$ = EXTRACT-MIN($Q$)
      $S = S \cup \{u\}$
          for each vertex $v \in Adj[u]$
              if $d[v] > d[u] + w(u, v)$
                  $d[v] = d[u] + w(u, v)$
                  $\pi[v] = u$
```

```
PARTITION($A$, $p$, $r$)
    $x = A[r]$
    $i = p - 1$
    for $j = p$ to $r - 1$
        if $A[j] \le x$
            $i = i + 1$
            exchange $A[i]$ with $A[j]$
    exchange $A[i + 1]$ with $A[r]$
    return $i + 1$
```

End of Question Paper