Sri Lanka Institute of Information Technology

B.Sc. Honours Degree in Information Technology

Specialized in Information Technology

Final Examination
Year 2, Semester 2 (2018)

IT2070-Data Structures and Algorithms

Duration: 2 Hours

October 2018

Instructions to Candidates:

- This paper is preceded by 10 minutes reading period. The supervisor will indicate when answering may commence.
- This paper has 4 questions.
- Answer questions in the booklet given.
- The total marks for the paper is 100.
- This paper contains 7 pages, including the cover page.
- Electronic devices capable of storing and retrieving text and mobile phones are not allowed.

Question 1 **(25 marks)**

a) Draw the resultant stack frame after performing the given operations to an empty stack s1 of size 4. (5 marks)

```
1) s1.push(6)
2) s1.push(10)
3) s1.push(11)
4) s1.push(s1.pop())
5) s1.push(1)
6) s1.peek()
```

b) i) A stack class called StackX has been created to store characters and 'push' and 'pop' methods have been implemented. Implement the peek method using push and pop methods. (4 marks)

ii) What is the time complexity of the peek method written in part b i) in Big O Notation ? (2 marks)

c) Complete the following five lines of the insert method of a circular queue class. (4 marks)

```
public void insert(int j) {
        if ............................... //line 1
            System.out.println("Queue is full");
        else {
                ........................   //line 2
                ........................ .. //line 3
                ........................   //line 4
                ........................ ... //line 5
                queArray[rear] = j;
                nItems++;
        }
}
```

d) *"rear = maxsize -1" can be used to find out whether a circular queue is full*

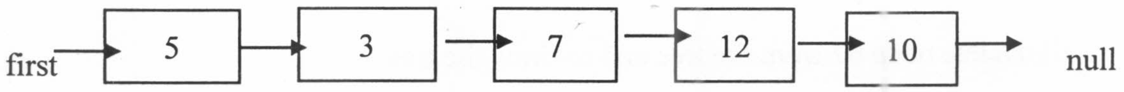
Do you agree with this statement? Justify your answer using a diagram (3 marks)

e) i) Explain how you would implement a queue using a linked list. (4 marks)

ii) Discuss one advantage of a queue implemented using a linked list over a queue implemented using an array. (3 marks)

Question 2 **(25 marks)**

a) Consider the following linked list.



Draw a diagram of the above linked list after the following lines of code have been executed. (2 marks)

```
Link temp = first.next;
Link nodeToInsert = new Link(4);
nodeToInsert.next = temp.next;
temp.next = nodeToInsert;
```

b) A linked list is used to store some employee data namely employee number (integer) and name (string).

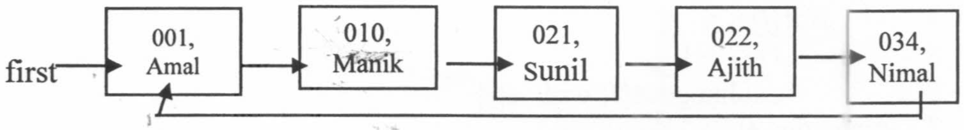
i) Implement **Link** class for the above linked list with suitable attributes and methods. (2 marks)

ii) Implement **Linked List** class with suitable attributes and the constructor. (1 mark)

iii) Write `sortedInsert(int empNo, String name)` method in **Linked List** class to insert the data into the linked list in ascending order. (6 marks)

iv) Write `displayList( )` method in **Linked List** class to display all the links in the linked list. (2 marks)

v) A singly circular linked list is a linked list where all the nodes are connected to each other. In circular linked list, the last link points to the first link as shown below.



Modify the `displayLink( )` method implemented in part b) iv) to print all the elements in the singly circular linked list.

(3 marks)

c) i) Insert the following numbers to a binary search tree. (2 marks)

56 34 23 67 38 88 60 58 59 25

ii) Delete node 67 from the tree and re-draw the tree (2 marks)

iii) Describe the type of the tree you get, if you first apply quick sort algorithm to the numbers given above and then create the tree? (2 marks)

iv) Discuss the drawback of the tree you get in part c) iii) , if there are any. (3 marks)

Question 3 (25 marks)

a) Find the ***Big O*** value for the following functions. Justify your answer.

a) $T(n) = n + n\log n + 100$

b) $T(n) = n^8 + n + 2^n + 1$ (2 marks)

b) Compare and contrast the Merge sort and Quick sort algorithms. (2 marks)

c) Given below is an algorithm for ***QUICKSORT*** *(A, p, r)*

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

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

i) Modify the above ***QUICKSORT*** *(A, p, r)* algorithm to sort the numbers in descending order. (Only the modified line should be described) (2 marks)

ii) Write a recurrence equation that describes the best case running time of quick sort algorithm. (4 marks)

d) Briefly describe the following terms as related to tree terminology.

i) Complete Binary tree
ii) Leaf Node

(3 marks)

e) Relationship between nodes (n) and height (h) of **Full Binary Tree** is given by $n = 2^{h+1} - 1$ Illustrate this using a diagram. (2 marks)

f) Illustrate the operations of the **Heap sort** for the array A of elements given below. (For the purpose of illustration, assign the values only once to the given algorithms at the end of the paper and use diagrammatic way to reach the answer.)

	1	2	3	4	5	6	7	8
A	25	15	35	55	5	75	65	45

(5 marks)

g) Consider the following **min-heap** A = <2, 7, 11, 16, 14, 12, 25, 40, 65>. Assume that the key 16 in the **min-heap** is modified to 40. Show the resulting **min-heap** after the modification.
min – heap property is **A[*PARENT*(*i*)] <= A[*i*]**

(5 marks)

Question 4 **(25 marks)**

a)

i. If modulo value is $q = 10$, how many spurious hits and valid hits do the **Rabin -Karp matcher** encounter in the text T = 6750100502007 when looking for pattern $P = 50$? (4 marks)

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

iii. What should be the number of spurious hits and valid hits if the best-case scenario occurs in Rabin-Karp algorithm? (2 marks)

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

(6 marks)

c) 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	a	b	a	a	a	a	b

Pattern ***P***

a	a	a

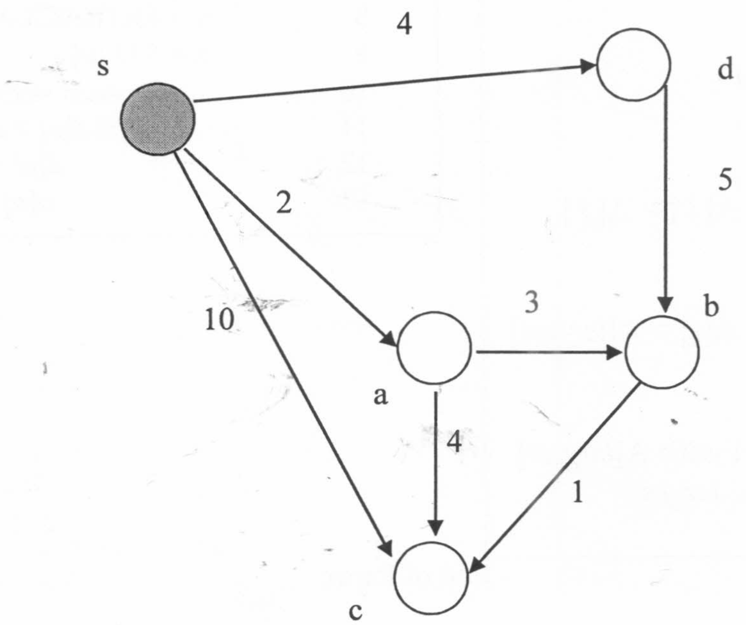
i) How many comparisons would occur in this algorithm? (1 mark)

ii) Show that the worst-case time complexity of the above algorithm is $O(m(n-m+1))$ where n is the number of characters in the text and m is the number of characters in the pattern.

(3 marks)

d) What is meant by the "**Minimum Cost Spanning Tree**" (MCST)? (2 marks)

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)



```
HEAP_INSERT(A, key)

A.heap_size = A.heap_size +1
i = A.heap_size
while i > 1 and A[PARENT(i)] < key
        A[i] = A[PARENT(i)]
        i = PARENT(i)
A[i] = key
```

```
HEAPSORT(A)
BUILD_HEAP[A]
for  i = A.length down to 2
    Exchange A[1] with A[i]
    A.heap_size = A.heap_size -1;
    HEAPIFY(A,1)


BUILD_HEAP (A)
A.heap_size = A.length
for i = ⌊A.length/2⌋ downto 1
        HEAPIFY(A,i)


HEAPIFY (A,i)
l = LEFT_CHILD (i);
r = RIGHT_CHILD (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]
        HEAPIFY (A, largest)
```

```
DIJKSTRA(G, w, s)

for each vertex v ∈ V[G]
        d[v] = ∞
        π[v] = NIL
d[s] = 0
S = Ø
Q = V[G]
while Q ≠ Ø
        u = EXTRACT-MIN(Q)
        S = S U {u}
            for each vertex v ∈ Adj[u]
                if d[v] > d[u] + w(u, v)
                    d[v] = d[u] + w(u, v)
                    π[v] = u
```

End of Paper