1

Code No11

FACULTY OF ENGINEERING

B.E III-Semester (CBCS) (I.T) (Suppl.) Examination, May / June 2019

Subject : Data Structures

Time : 3 Hours   Max Marks : 70

Note: Answer all questions from Part – A & Any five questions from Part – B.

Part - A (20 Marks)

1. Define the terms Space and Time complexity.
2. What is a template? Give an example.
3. Define Abstract data type and Write ADT for Queue.
4. Convert the following infix expression to prefix and postfix
   (A * B * (C – A) – D)
5. Explain differences between linear and linked representation of a stack.
6. Define a double linked list with an example.
7. Why hashing is needed? What are its advantage over others?
8. Differentiate between trees and graphs.
9. Evaluate the following postfix expression 5 6 2 + * 12 4 /
10. Give the worst, best and average complexities for merge and selection sort.

Part-B (20 Marks)

11. a) Write a C++ program for implementing Insertion and Deletion in string ADT.   (5)

    b) Compute the best and worst case step count analysis for the following function

       int sequential_search(int *a , const int n, const int x)   (5)

          { int i ;

          for(i=0; i< n && a[i]!=x; i++);

          if (i = = n) return -1;

          else return i ; }

12. Write a C++ code to implement stack as an ADT using templates.   (10)

13. Write procedure for insertion and deletion of an element from a doubly linked list   (10)

14. What is Minimum cost spanning tree. Use Prim's algorithm to find minimum cost

spanning tree for the following graph.   (10)

...2
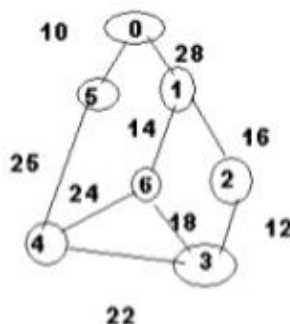
15. a) Construct AVL for the following input sequence:

   15, 6, 25, 11, 10, 13, 3, 29, 37 (6)

   b) Define Max-Heap. Explain how to insert an element into a Max Heap. (4)
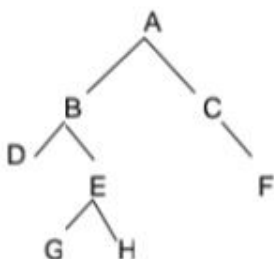
16. a) Explain Merge sort for the following elements.

   66, 48, 57, 92, 24, 65, 83, 72 (6)

   b) Write In order, Preorder, Post order and level order for the following binary Search tree. (4)

17. Write short notes on any of two: (10)

   a) Asymptotic notation.

   b) Red Black Tree.

   c) Quick Sort.

....

Code No. 15049 / AICTE

FACULTY OF ENGINEERING

B.E. (I.T) III – Semester (AICTE) (Main & Backlog) Examination, July 2021

Subject: Data Structures

Time: 2 hours   Max. Marks: 70

Note: Missing data, if any may be suitably assumed.

PART – A

(5x2 = 10 Marks)

Answer any five questions.

1. Define Data Structure. Write the characteristics of Data Structures.
2. Write an ADT for an Array.
3. Define single, double and circular linked lists.
4. Write the postfix expression of $A + B * (C – D) / (E – F)$.
5. State the difference between complete binary tree and full binary tree.
6. When is an undirected graph said to be 'connected'?
7. Give the complexities for the following sorting algorithms
   - a) Insertion sort   b) Merge sort   c) Quick sort   d) Heap sort
8. Define polymorphism and function overloading.
9. Write about Threaded Binary Tree with an example.
10. What is Hash Function? List few Hash Function.

PART – B

(4x15 = 60 Marks)

Answer any four questions.

11. Explain about space complexity and time complexity. Evaluate the time complexity for the following iterative function.

| Line No.	Float sum (float * a, const int n)
	{
1	Float s = 0
2	for (int i = 0 ; i < n ; i++)
3	S += a[i] ;
4	return s;
5	}
6	

12. Write a C++ code to implement following operations on stack.
     a) Push   b) Pop   c) Display   d) Top

13. Define BST. Create a binary search tree with the following keys and perform in order, pre-order, post-order traversals on it.
30, 20, 25, 40, 35, 36, 32, 45, 42

..2

---

14. Explain prim's algorithm and find minimum cost spanning tree for the following graph.
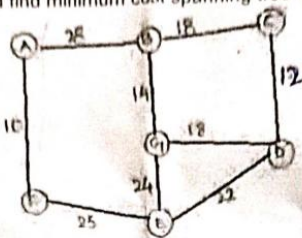
15. Explain the working of quick sort. Sort the following sequence of keys using quick sort 66, 77, 11, 88, 99, 22, 33, 44, 55. Show different passes (Trace) indicating the pivot and the partitions formed specify its time complexity.

16. Write an algorithm to add and subtract two polynomials using linked list.

17. Write short notes on following:
     a) Collision handling techniques in hashing
     b) AVL Trees

****

Code No. 2914 / AICTE

FACULTY OF ENGINEERING

B.E. (I.T) (AICTE) III – Semester (Main) Examination, December 2019

Subject: Data Structures

Time: 3 Hours

Max.Marks: 70

Note: Answer all questions from Part-A and any five questions from Part-B

PART – A (10x2 = 20 Marks)

1. What are access modifiers?
2. What is a class and object?
3. Explain program organisation in C++?
4. Differentiate between array and linked list.
5. Convert the following infix expression to postfix form A+B/C*D-E
6. Write the difference between graphs and trees.
7. What are threaded binary trees?
8. What is a balance factor of a node in AVL tree?
9. What is spanning tree and minimum cost spanning tree?
10. What is weighted graph? Give example.

PART – B (5x10 = 50 Marks)

11 a) Explain functions and inline functions in C++ with examples and how they are different? (6)
   b) Write a recursive program in C++ to find the factorial of a given number. (4)

12 a) Write a C++ program for stack as ADT? (5)
   b) Explain inheritance methods in C++. (5)

13 What are templates in C++? Explain two types of templates with examples. (10)

14 a) What is hashing? (2)
   b) Explain the following over flow handling techniques linear probing, quadratic probing and chaining. (8)

15 Create a AVL tree in following order MARCH, MAY, NOVEMBER, AUGUST, APRIL, JANUARY, DECEMBER, JULY, FEBRUARY, JUNE, OCTOBER, SEPTEMBER (based on alphabetical order i.e., January > February because j > f August > April because second letter u > p). (10)

16 a) Explain BFS with an example. (5)
   b) Write Prim's algorithm and explain with an example. (5)

17 Write short notes on:
   a) Merge sort (4)
   b) Max heap and min heap (3)
   c) Binary tree traversal techniques. (3)

****

FACULTY OF INFORMATICS

B.E (I.T) III-Semester (CBCS) (Supply) Examination, May/June 2018

Subject : Data Structures

Time : 3 Hours   Max Marks : 70

Note: Answer all questions from Part – A & Any five questions from Part – B. Part - A (20

Marks)

1. What is sparse matrix? How sparse matrix is represented.
2. Define Abstract data type and Write ADT for Stack.
3. Transform the following expression to prefix and postfix form. (A / B *
   C * D+E)
4. Write a function to check whether circular queue is full or not.
5. Which is more efficient in terms of running time, the arrays or linked implementation of
   queue? Why?
6. What is hash function? List few hash functions.
7. Explain the array and linked representation of Binary Tree.
8. Compare and contrast DFS and BFS.
9. For insertion sort, what arrangement of the input data will cause the worst case
   performance? The best case performance?
10. Define splay tree.

Part-B (50 Marks)

11.a) Explain various asymptotic notations with examples.   5

   b) Write a C++ function to reverse a given string.   5

12. Write an algorithm for conversion of an infix expression to postfix expression and trace
    the algorithm for the expression ((((A/B)*C)*D) + E) and get the resultant
    postfix expression   10

13. Write C++ code to implement various operation on queue (use linked representation) 10

14. What is minimum cost spanning tree. Use Kruskal's algorithm to find minimum cost

    spanning tree for the following graph.

10   0   28
5   1

the algorithm for the expression ((((A/B)*C)*D) + E) and get the resultant
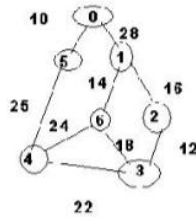postfix expression   10

13. Write C++ code to implement various operation on queue (use linked representation) 10

14. What is minimum cost spanning tree. Use Kruskal's algorithm to find minimum cost

spanning tree for the following graph.   10
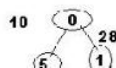
Code No. 458/CBCS/S

-2-

15. Write and explain heap sort algorithm. And Construct a min heap for the following
    elements   125, 145, 42, 55, 88, 72, 62, 95, 25, 100   10

16.a) Make a Binary Search Tree for the following sequence of numbers

       {30, 20, 25, 40, 35, 36, 32, 45, 42} Traverse the obtained BST in Pre-order

       Post-order and In-order.   5

    b) Write about different graph representations. Use Examples   5

17. Write short notes on any of two: 10

    a) Threaded binary tree

    b) Secure hash function

    c) AVL Trees.

****