DATA STRUCTURES THROUGH C++

Subject Code: CS302ES
Regulations : R16 - JNTUH
Class : II Year B.Tech CSE I Semester

Department of Computer Science and Engineering

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

Ibrahimpatnam - 501 510, Hyderabad

DATA STRUCTURES THROUGH C++ [CS302ES]
COURSE PLANNER

I. COURSE OVERVIEW:

Basic principles and techniques for Data structures. Students will gain experience in how to keep a data in an ordered fashion in the computer. Students can improve their programming skills using Data Structures Concepts through C++ .

II. PREREQUISITE:

1. C programming language

III. COURSE OBJECTIVE:

S. No	Objective
1	Introduce the student to the concept of data structures through abstract data structures including lists, sorted lists, stacks, queues, deques, sets/maps, directed acyclic graphs, and graphs; and implementations including the use of linked lists, arrays, binary search trees, *M*-way search trees, hash tables, complete trees, and adjacency matrices and lists.
2	Introduce the student to algorithms design including greedy, divide-and-conquer, random and backtracking algorithms and dynamic programming; and specific algorithms including, for example, resizing arrays, balancing search trees, shortest path, and spanning trees.

IV. COURSE OUTCOME:

S.No	Description	Bloom's Taxonomy Level
1	Understand numerous examples of relationships between data;	**Knowledge, Understand** **(Level1, Level2)**
2	Understand the purpose and mathematical background of algorithm analysis and be able to apply this to determine the run time and memory usage of algorithms;	**Apply, Create** **(Level 3)**
3	Understand the variety of ways that linearly and weakly ordered data can be stored, accessed, and manipulated;	**Evaluate** **(Level 3)**
4	Understand various sorting algorithms and the run-time analysis required to determine their efficiencies;	**Analyze** **(Level 3)**
5	Understand numerous algorithm design techniques including greedy, divide-and-conquer, dynamic programming, randomized algorithms, and backtracking;	**Apply (Level 3)**

V. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes (PO)		Level	Proficiency assessed by
PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	3	Assignments, Tutorials, Mock Tests
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems	3	Assignments, Tutorials, Mock

Program Outcomes (PO)		Level	Proficiency assessed by
	reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.		Tests
PO3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	3	Assignments, Tutorials, Mock Tests
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	-	
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	-	
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.	3	Assignments, Tutorials, Mock Tests --
PO7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	2	Assignments, Tutorials,
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	1	Assignments
PO9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	-	--
PO10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	-	--
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	-	--
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological	2	Assignments, Tutorials

Program Outcomes (PO)		Level	Proficiency assessed by
	change.		

1: Slight (Low)	2: Moderate (Medium)	3: Substantial (High)	- : None

VI. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes (PSO)		Level	Proficiency assessed by
PSO1	**Software Development and Research Ability:** Ability to understand the structure and development methodologies of software systems. Possess professional skills and knowledge of software design process. Familiarity and practical competence with a broad range of programming language and open source platforms. Use knowledge in various domains to identify research gaps and hence to provide solution to new ideas and innovations.	3	Lectures, Assignments, Tutorials, Mock Tests
PSO2	**Foundation of mathematical concepts:** Ability to apply the acquired knowledge of basic skills, principles of computing, mathematical foundations, algorithmic principles, modeling and design of computer- based systems in solving real world engineering Problems.	3	Lectures, Assignments, Tutorials, Mock Tests
PSO3	**Successful Career:** Ability to update knowledge continuously in the tools like Rational Rose, MATLAB, Argo UML, R Language and technologies like Storage, Computing, Communication to meet the industry requirements in creating innovative career paths for immediate employment and for higher studies.	2	Lectures, Assignments
1: Slight (Low)	**2: Moderate (Medium)**	**3: Substantial (High)**	**- : None**

VII. SYLLABUS:

UNIT - I

C++ Programming Concepts: Review of C, input and output in C++, functions in C++ value parameters, reference parameters, Parameter passing, function overloading, function templates, Exceptions-throwing an exception and handling an exception, arrays, pointers, new and delete operators, class and object, access specifiers, friend functions, constructors and destructor, Operator overloading, class templates, Inheritance and Polymorphism..

Basic Concepts - Data objects and Structures, Algorithm Specification-Introduction, Recursive algorithms, Data Abstraction, Performance analysis- time complexity and space complexity, Asymptotic Notation-Big O, Omega and Theta notations, Complexity Analysis Examples, Introduction to Linear and Non Linear data structures.

UNIT - II

Representation of single, two dimensional arrays, sparse matrices-array and linked representations. Linear list ADT-array representation and linked representation, Singly Linked Lists- Operations-Insertion, Deletion, Circularly linked lists-Operations for Circularly linked lists, Doubly Linked Lists- Operations- Insertion, Deletion. Stack ADT, definition, array and linked implementations, applications-infix to postfix conversion, Postfix expression evaluation, recursion implementation, Queue ADT, definition, array and linked Implementations, Circular queues-Insertion and deletion operations

UNIT - III

Trees – definition, terminology, Binary trees-definition, Properties of Binary Trees, Binary Tree ADT, representation of Binary Trees-array and linked representations, Binary Tree traversals, Threaded binary trees, Priority Queues –Definition and applications, Max Priority Queue ADT-implementation-Max Heap-Definition, Insertion into a Max Heap, Deletion from a Max Heap.

UNIT - IV

Searching - Linear Search, Binary Search, Hashing-Introduction, hash tables, hash functions, Overflow Handling, Comparison of Searching methods. Sorting-Insertion Sort, Selection Sort, Radix Sort, Quick sort, Heap Sort, Merge sort, Comparison of Sorting methods.

UNIT - V

Graphs–Definitions, Terminology, Applications and more definitions, Properties, Graph ADT, Graph Representations- Adjacency matrix, Adjacency lists, Graph Search methods - DFS and BFS, Complexity analysis, Search Trees-Binary Search Tree ADT, Definition, Operations- Searching, Insertion and

Deletion, Balanced search trees-AVL Trees-Definition and Examples only, B-Trees-Definition and Examples only, Red-Black Trees-Definitions and Examples only, Comparison of Search Trees.

TEXT BOOKS:

1. Data structures, Algorithms and Applications in C++, 2nd Edition, Sartaj Sahni, Universities Press.
2. Data structures and Algorithms in C++, Adam Drozdek, 4th edition, Cengage learning.

REFERENCE BOOKS:

1. Data structures with C++, J. Hubbard, Schaum’s outlines, TMH.
2. Data structures and Algorithms in C++, M.T. Goodrich, R. Tamassia and D. Mount, Wiley India.
3. Data structures and Algorithm Analysis in C++, 3rd edition, M. A. Weiss, Pearson.
4. Classic Data Structures, D. Samanta, 2nd edition, PHI.

RELEVANT SYLLABUS FOR GATE:

structured programming with Pascal/C including recursion; arrays, stacks, strings, queues, lists, trees, sets and graphs; algorithm for tree and graphs traversals, connected component, spanning trees, shortest paths; hashing, sorting and searching algorithm design and analysis techniques, big 'oh' notation, solution of sample recurrence relations.

RELEVANT SYLLABUS FOR IES:

-NA-

VIII. COURSE PLAN (WEEK-WISE):

Session	Week	Topic	Course Learning Outcomes	Reference
Unit I				2, 3
1.	1	Review of C, input and output in C++, functions in C++ value parameters, reference parameters	**Gathering** the Ideas about the basic and the higher C++ Information	
2.		Parameter passing, function overloading, function templates	**Define** Parameter Passing	
3.		Parameter passing, function overloading, function templates	**Code** the Parameter Passing Mechanisms	
4.		Exceptions-throwing an exception and handling an exception		
5.		arrays, pointers, new and delete operators, class and object, access specifiers,		
6.	2	friend functions, constructors and destructor	**Synthesis** the idea of followings: Friend function, Operator overloading, Recursive Algorithms	
7.		Operator overloading, class templates		
8.		Inheritance and Polymorphism		
9.		Basic concepts- Data objects and Structures, Algorithm Specification-Introduction		
10.		Recursive algorithms		
11.	3	Data Abstraction, Performance analysis- time complexity and space complexity	**Synthesis** the idea of followings: Data Abstraction, Asymptotic Notation – Big O, Omega and Theta	
12.		Asymptotic Notation-Big O, Omega and Theta notations		
13.		Asymptotic Notation-Big O, Omega and Theta notations		
14.		Introduction to Linear and Non Linear data structures		
15.		*Mock Test #1*		
Unit II				
16.	4	Representation of single, two dimensional arrays	**Define** Single and multidimensional array	
17.		sparse matrices-array and linked representations	**Define** Sparse Matrices	
18.		Linear list ADT, Array and Linked representation	**Define** Linear ADT, Array, Linked List	
19.		Singly Linked Lists-Operations-Insertion, Deletion, Concatenating singly linked lists		
20.		*Bridge Class #1 / Seminar / Guest Lecture*		

21.	5	Singly Linked Lists-Operations-Insertion, Deletion	**Synthesis** Linked List Programming's	
22.		Circularly linked lists- Operations for Circularly linked lists		
23.		Doubly Linked Lists- Operations- Insertion, Deletion		
24.		Stack ADT, definition, operations		
25.		*Bridge Class #2 / Seminar / Guest Lecture*		
26.	6	array and linked implementations in C	**Gathering** the **Knowledge** about Array usages in using C and C++	
27.		applications-infix to postfix conversion		
28.		Postfix expression evaluation		
29.		recursion implementation		
30.		*Bridge Class #3 / Seminar / Guest Lecture*		
31.	7	Queue ADT, definition and operations ,array and linked Implementations in C,	**Synthesis** Queue ADT with array and Linked List	
32.		Queue ADT, definition and operations ,array and linked Implementations in C,		
33.		Circular queues-Insertion and deletion operations		
34.		Circular queues-Insertion and deletion operations		
35.		*Bridge Class #4 / Seminar / Guest Lecture*		
Unit III				
36.	8	Trees – Definition, Terminology, Representation of Trees	**Synthesis** the Trees and Binary Tree ADTs with the characteristics.	
37.		Binary tree – Definition, Binary tree ADT		
38.		Properties of Binary Trees		
39.		Binary Tree Representations-array and linked representations		
40.		*Bridge Class #5 / Seminar*		
	Mid-Term #1 Examinations (Week 9)			
41.	10	Binary Tree Representations-array and linked representations	**Gathering** the Knowledge about the Tree traversal and **Improve** the Coding Skills	
42.		Binary Tree traversals		
43.		Threaded binary trees		
44.		Priority Queue – Definition and Applications		
45.		*Bridge Class #6 / Seminar*		
46.	11	Max Priority Queue ADT-implementation		
47.		Max Priority Queue ADT-implementation		
48.		Max Heap-Definition, Insertion into a Max Heap		
49.		Deletion from a Max Heap		
50.		*Bridge Class #7/ Seminar / Guest Lecture*		
Unit IV				1, 2
51.	12	Searching- Linear Search	**Gathering** the knowledge about all the searching and sorting algorithms	
52.		Binary Search		
53.		Static Hashing-Introduction		
54.		hash tables		
55.		*Bridge Class #8 / Seminar*		
56.	13	hash functions		
57.		Overflow Handling		
58.		Sorting- Insertion Sort,		
59.		Selection Sort		
60.		*Bridge Class #9 / Seminar*		

61.	14	Radix Sort, Merge sort	**Gathering** the knowledge about all the searching and sorting algorithms	
62.		Quick sort		
63.		Heap Sort		
64.		Comparison of Sorting methods		
65.		*Mock Test #2*		
Unit V				
66.	15	Graphs – Introduction, Definition, Terminology, Applications and Properties	**Gathering** the knowledge about Graph techniques and improve the coding skills using graph	
67.		Graph ADT, Graph Representations- Adjacency matrix, Adjacency lists		
68.		Graph traversals- DFS		
69.		Graph traversals- BFS, Complexity analysis		
70.		*Bridge Class #10 / Seminar / Guest Lecture*		
71.	16	Search Trees-Binary Search Trees ADT, Definition	**Gathering** the knowledge about Search tree and binary Tree techniques and improve the coding skills	
72.		Operations- Searching, Insertion and Deletion		
73.		AVL Trees-Definition and Examples		
74.		Insertion into an AVL Tree		
75.		*Bridge Class #11 / Seminar*		
76.	17	B-Trees, Definition, B-Tree of order m, Examples	**Gathering** the knowledge about B - Tree and Red Black Splay Trees techniques and improve the coding skills	
77.		operations-Insertion and Searching		
78.		Introduction to Red-Black and Splay Trees(Elementary treatment-only Definitions and Examples)		
79.		Comparison of Search Trees		
80.		*Bridge Class #12 / Seminar*		
	Mid-Term #2 Examinations (Week 18)			

IX. MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

	Program Outcomes												Program Specific Outcomes		
	P O 1	P O 2	P O 3	P O 4	P O 5	P O 6	P O 7	P O 8	P O 9	PO 10	PO 11	PO 12	PS O 1	PS O 2	PS O 3
CO 1	1	1	2	-	-	-	-	-	-	-	-	-	1	-	1
CO 2	2	2	-	-	-	-	-	-	-	-	-	-	1	2	2
CO 3	2	-	3	-	3	-	-	-	-	-	-	-	2	-	2
CO 4	2	2	-	2	-	3	-	-	-	-	-	-	-	-	-
CO 5	1	2	-	-	-	-	-	-	-	-	-	-	1	-	2
AV G	1.6	1.4	1.0	0.4	0.6	0.6	-	-	-	-	-	-	1	0.4	1.4

X. QUESTION BANK (JNTUH)

UNIT I

Long Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Describe the concept of function overloading with an example.	Knowledge	1
2	What is meant by Template? Explain with an example class templates and function templates.	Understand	2
3	What is meant by time complexity? Define different time complexity notations. Give examples one for each.	Knowledge	1
4	Write a short note on constructor and destructor and give examples	Knowledge	1
5	Describe in detail about friend functions	Knowledge	1

Short Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Differentiate between new and delete operator	Knowledge	2
2	Explain in brief class and object, access specifiers	Knowledge	2
3	Differentiate between linear and non linear data structures	Knowledge	2

UNIT II

Long Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Explain about sparse matrices	Understanding	3

2	What is Stack? Explain about application of stack?	Understanding	3
3	Explain about operations in single linked list	Understanding	3
4	Explain about operations in circular linked list.	Understanding	3
5	What is Queue? Define the implementation with array and linked list?	Understanding	3

Short Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Differentiate between Single linked list and circular linked list	Synthesis	1, 3
2	Differentiate between Double linked list and single linked list	Synthesis	1,3
3	Differentiate between Double linked list and circular linked list	Synthesis	1
4	Differentiate between Queue and Double ended queue	Synthesis	1

UNIT III

Long Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	What is Binary Tree? Explain about operations on Binary tree?	Define	4
2	Describe in brief about array and linked representations of binary tree	Understand	3
3	Write a short note on Threaded binary trees	Knowledge	5
4	Describe the concept of tree traversals with an example.	Synthesis	2
5	Describe a procedure to insert and delete an element into a Max Heap	Synthesis	3
6	Differentiate between tree and binary tree	Synthesis	3

Short Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Define threaded binary tree and give an example	Understand	4
2	Define the following a. Max heap b. Min heap	Knowledge	3
3	What is a priority queue? Mention the applications of priority queue	Synthesis	5

UNIT IV

Long Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	What is searching? Explain Binary and Linear Search?	Understand	4
2	What is sorting? Explain about Selection and Heap	Understand	3

	Sorting		
3	Illustrate the concept of Merge sort and Quick sort	Understand	5
4	Implement Quick sort using C++	Knowledge	4
5	Explain Insertion and Radix Sort?	Knowledge	3

Short Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Differentiate between linear search and binary search	Understand	4
2	Differentiate between quick sort and merge sort	Understand	3
3	What is the time complexity of quick sort in best, worst and average case	Understand	5
4	Define hash tables	Knowledge	4
5	Write an algorithm of Merge sort	Knowledge	3

UNIT V

Long Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Write a short note on representation of Graphs	Understand	2
2	Describe a procedure about insertion and deletion operations of BST	Understand	3
3	Explain about B-trees? Define The Operations on B-Tree?	Understand	4
4	Explain about Red-Black and Splay trees?	Understand	2
5	Explain the operations on AVL Tree?	Analysis	5

Short Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	What is Graph? Explain Graph Traversal Techniques?	Knowledge	2
2	Differentiate between graph and tree	Knowledge	3
3	Differentiate between BST and AVL tree	Analysis	4
4	Define AVL tree. what is the acceptable balancing factor of AVL tree	Analysis	2

OBJECTIVE QUESTIONS:

UNIT I

1. Which of the following is not a type of constructor?

A.Copy constructor
B.Friend constructor
C.Default constructor
D.Parameterized constructor
Answer: Option B

2. Which of the following is not the member of class?

A. Static function
B. Friend function
C. Const function
D. Virtual function
Answer: Option B

3. Which of the following term is used for a function defined inside a class?

A. Member Variable
B. Member function
C. Class function
D. Classic function
Answer: Option B

4. Which of the following concepts of OOPS means exposing only necessary information to client?

A. Encapsulation
B. Abstraction
C. Data hiding
D. Data binding
Answer: Option C

5. Which of the following statement is correct?

A. A constructor is called at the time of declaration of an object.
B. A constructor is called at the time of use of an object.
C. A constructor is called at the time of declaration of a class.
D. A constructor is called at the time of use of a class.

Answer: Option A

6. Which of the following is correct about function overloading?

A. The types of arguments are different.
B. The order of argument is different.
C. The number of argument is same.
D. Both A and B.

Answer: Option D

UNIT II

1. One difference between a queue and a stack is:
A. Queues require linked lists, but stacks do not.
B. Stacks require linked lists, but queues do not.
C. Queues use two ends of the structure; stacks use only one.
D. Stacks use two ends of the structure, queues use only one.

2. If the characters 'D', 'C', 'B', 'A' are placed in a queue (in that order), and then removed one at a time, in what order will they be removed?
   A. ABCD
   B. ABDC
   C. DCAB
   D. DCBA

3. Consider the implementation of the Queue using a circular array. What goes wrong if we try to keep all the items at the front of a partially-filled array (so that data[0] is always the front).
   A. The constructor would require linear time.
   B. The remove method would require linear time.
   C. The insert method would require linear time.
   D. The is Empty method would require linear time.

4. In the circular array version of the Queue class, which operations require linear time for their worst-case behavior?
   A. remove
   B. insert when the capacity has not yet been reached
   C. is Empty
   D. None of these operations require linear time

5. Time complexity of binary search algorithm is
   a)n b)nlogn c)logn d)n^2

6. Sprase matrix have
   a) many zero entries b) many non-zero entries
   c) higher dimension d) none of the above

7. .... form of access is used to add and remove nodes from a queue.
   A. LIFO, Last In First Out B. FIFO, First In First Out
   C. Both a and b D. None of these

8. In linked representation of stack the null pointer of the last node in the list signals ....
   A. Beginning of the stack B. Bottom of the stack
   C. Middle of the stack D. In between some value

9. Which of the following name does not relate to stacks?
   A. FIFO lists B. LIFO lists
   C. Piles D. Push down lists

10. Which of the following is an application of stack?
   A. finding factorial B. tower of Hanoi
   C. infix to postfix D. all of the above

UNIT III

1. The height of a BST is given as h. Consider the height of the tree as the no. of edges in the longest path from root to the leaf. The maximum no. of nodes possible in the tree is?
   a)2^{h-1}-1 b)2^{h+1}-1
   c)2^h+1 d) 2^{h-1} +1 ANSWER : B

2. The no of external nodes in a full binary tree with n internal nodes is?
   a) n b) n+1
   c) 2n d) 2n + 1 ANSWER: B

3. Suppose a binary tree is constructed with n nodes, such that each node has exactly either zero or two children. The maximum height of the tree will be?
   a) (n+1)/2 b) (n-1)/2
   c) n/2 -1 d) (n+1)/2 -1 ANSWER: B
4. Which of the following statement about binary tree is CORRECT?
   a) Every binary tree is either complete or full
   b) Every complete binary tree is also a full binary tree
   c) Every full binary tree is also a complete binary tree
   d) A binary tree cannot be both complete and full

ANSWER: C

5. Suppose we have numbers between 1 and 1000 in a binary search tree and want to search for the number 363. Which of the following sequence could not be the sequence of the node examined?
   a) 2, 252, 401, 398, 330, 344, 397, 363 b) 924, 220, 911, 244, 898, 258, 362, 363
   c) 925, 202, 911, 240, 912, 245, 258, 363 d) 2, 399, 387, 219, 266, 382, 381, 278, 363
   ANSWER: C
6. **6.** In full binary search tree every internal node has exactly two children. If there are 100 leaf nodes in the tree, how many internal nodes are there in the tree?
   a) 25 b) 49 c) 99 d) 101
   ANSWER: C
7. Which type of traversal of binary search tree outputs the value in sorted order?
   a) Pre-order b) In-order c) Post-order d) None
   ANSWER: B
8. **8.** Suppose a complete binary tree has height h>0. The minimum no of leaf nodes possible in term of h is?
   a) 2^h -1 b) 2^{h-1} + 1 c) 2^{h-1} d) 2^h +1
   ANSWER: C

UNIT IV

1) The worst case occur in linear search algorithm when ....
   A. Item is somewhere in the middle of the array
   B. Item is not in the array at all
   C. Item is the last element in the array
   D. Item is the last element in the array or item is not there at all

   ANSWER: D
2) If the number of records to be sorted is small, then ...... sorting can be efficient.
   A. Merge
   B. Heap
   C. Selection
   D. Bubble

   ANSWER: C
3) The complexity of sorting algorithm measures the ...... as a function of the number n of items to be sorter.
   A. average time

B. running time
C. average-case complexity
D. case-complexity
ANSWER: B

4) Which of the following is not a limitation of binary search algorithm?
A. must use a sorted array B. requirement of sorted array is expensive when a lot of insertion and deletions are needed
C. there must be a mechanism to access middle element directly
D. binary search algorithm is not efficient when the data elements more than 1500.
ANSWER: D

5) The Average case occurs in linear search algorithm ....
A. when item is somewhere in the middle of the array
B. when item is not the array at all
C. when item is the last element in the array
D. Item is the last element in the array or item is not there at all
ANSWER: A

6) Binary search algorithm cannot be applied to ...
A. sorted linked list
B. sorted binary trees
C. sorted linear array
D. pointer array
ANSWER: A

7) Complexity of linear search algorithm is ....
A. O(n)
B. O(logn)
C. O(n2)
D. O(n logn)
ANSWER: A

8) Sorting algorithm can be characterized as ......
A. Simple algorithm which require the order of n2 comparisons to sort n items.
B. Sophisticated algorithms that require the O(nlog2n) comparisons to sort items.
C. Both of the above
D. None of the above
ANSER: C

9) The complexity of bubble sort algorithm is .....
A. O(n)
B. O(logn)
C. O(n2)
D. O(n logn)
ANSWER: C

10) State True or False for internal sorting algorithms.
i) Internal sorting are applied when the entire collection if data to be sorted is small enough that the sorting can take place within main memory.
ii) The time required to read or write is considered to be significant in evaluating the performance of internal sorting.
A. i-True, ii-True B. i-True, ii-False
C. i-False, ii-True D. i-False, ii-False
ANSWER: B

11) The complexity of merge sort algorithm is ......
A. O(n) B. O(logn) C. O(n2) D. O(n logn)

ANSWER: D

12) .... is putting an element in the appropriate place in a sorted list yields a larger sorted order list.

A. Insertion
B. Extraction
C. Selection
D. Distribution

ANSWER: A

13) ....order is the best possible for array sorting algorithm which sorts n item.

A. O(n logn)
B. O(n2)
C. O(n+logn)
D. O(logn)

ANSWER: C

14) .... is rearranging pairs of elements which are out of order, until no such pairs remain.

A. Insertion
B. Exchange
C. Selection
D. Distribution

ANSWER: B

15) .... is the method used by card sorter.

A. Radix sort B. Insertion C. Heap D. Quick

ANSWER: A

UNIT V

1) In a graph if e=(u,v) means ....

A. u is adjacent to v but v is not adjacent to u. B. e begins at u and ends at v
C. u is node and v is an edge. D. both u and v are edges.

2) Which of the following is AVL Tree?

A

```
        100
     /      \
    50       200
   /            \
 10             300
```

B

```
           100
       /       \
     50         200
    /        /     \
  10       150      300
 /
5
```

C

```
            100
       /          \
     50            200
    /  \          /     \
  10    60       150     300
 /                 \        \
5                   180       400
```

A. only A B. only A and C C. A,B and C D. only B

3) Which of the following is a self-adjusting or self-balancing Binary Search Tree

A Splay Tree

B AVL Tree

C Red Black Tree

D All the above

4. Which of the following operations are used by Red-Black trees to maintain balance during insertion/deletion?
a) Recoloring of nodes
b) Rotation (Left and Right)
A. only A B. only B C. both A and B D. none

5. The balance factor of every node in an AVL tree may be _________.

1. 0
2. 1
3. -1
4. 2

A. 1,2,3 B. 2,3,4 C. 1,3,4 D. 1,2,4

QUESTIONS RELATED TO THE SUBJECT IN GATE:

1. The pre order traversal sequence of a binary search tree is 30, 20, 10, 15, 25, 23, 39, 35, 42. Which one of the following is the post order traversal sequence of the same tree?
(A) 10, 20, 15, 23, 25, 35, 42, 39, 30 (B) 15, 10, 25, 23, 20, 42, 35, 39, 30
(C) 15, 20, 10, 23, 25, 42, 35, 39, 30 (D) 15, 10, 23, 25, 20, 35, 42, 39, 30

2. Suppose a circular queue of capacity (n −1) elements is implemented with an array of n elements.
Assume that the insertion and deletion operations are carried out using REAR and FRONT as array index variables, respectively. Initially, REAR = FRONT = 0. The conditions to detect *queue full* and *queue empty* are
(A) *full*: (REAR+1) mod n == FRONT
empty: REAR == FRONT
(B) *full*: (REAR+1) mod n == FRONT
empty: (FRONT+1) mod n == REAR
(C) *full*: REAR == FRONT
empty: (REAR+1) mod n == FRONT
(D) *full*: (FRONT+1) mod n == REAR

empty: REAR == FRONT

3. The height of a tree is defined as the number of edges on the longest path in the tree. The function Shown in the pseudo code below is invoked as height(root) to compute the height of a binary tree rooted at the tree pointer root.

```
int height (tree ptr n){ if (n == NULL) return -1;
if (n  left == NULL)
if (n  right == NULL) return 0;
else return ; // Box 1
else { h1 = height (n  left);
if (n  right == NULL) return (1+h1);
else { h2 = height (n  right);
return ; // Box 2
}
}
}
```

The appropriate expressions for the two boxes B1 and B2 are

(A) B1: (1+height(n right))
B2: (1+max(h1, h2))

(B) B1: (height(n right))
B2: (1+max(h1,h2))

(C) B1: height(n right)
B2: max(h1, h2)

(D) B1: (1+ height(n right))

4. Two main measures for the efficiency of an algorithm are

a. Processor and memory
b. Complexity and capacity
c. Time and space
d. Data and space

5. The time factor when determining the efficiency of algorithm is measured by

a. Counting microsecond’s
b. counting the number of key operations
c. counting the number of statements
d. counting the kilobytes of algorithm

6. The space factor when determining the efficiency of algorithm is measured by

a. Counting the maximum memory needed by the algorithm
b. counting the minimum memory needed by the algorithm
c. counting the average memory needed by the algorithm
d. counting the maximum disk space needed by the algorithm

7. Which of the following case does not exist in complexity theory?

a. Best case
b. Worst case
c. Average case
d. Null case

8. The Worst case occur in linear search algorithm when

a. Item is somewhere in the middle of the array
b. Item is not in the array at all
c. Item is the last element in the array
d. Item is the last element in the array or is not there at all

9. The Average case occur in linear search algorithm
a. When Item is somewhere in the middle of the array
b. When Item is not in the array at all
c. When Item is the last element in the array
d. When Item is the last element in the array or is not there at all

IES QUESTIONS:

Not Applicable

WEBSITES' ADDRESSES:

1. http://www.dreamincode.net/forums/forum/48-c-tutorials/
2. http://nptel.iitm.ac.in/video.php?subjectId=106102064
3. http://www.tutorialspoint.com/cplusplus/cpp_data_structures.htm
4. http://www.sourcecodesworld.com/source/BrowseCategory.asp?CatId=33

EXPERT DETAILS:

1. Dr. Naveen Garg from IIT DELHI.
2. Dr.Pradip Das from IIT Guwahati
3. Dr.Padmanabam from JNTUH.

LIST OF TOPICS FOR STUDENTS' SEMINARS:

1. Applications of Trees
2. Comparative study of all the data structures.
3. Applications of Graphs.
4. Comparative study of all the types of trees.

CASE STUDIES / SMALL PROJECTS:

Implement the following programs using C++

1. Concatenation of two Single Linked List
2. Removing duplicate element of linked list
3. Queue using two stacks
4. Splay trees