NGM COLLEGE (AUTONOMOUS) POLLACHI

DEPARTMENT OF INFORMATION TECHNOLOGY

18UIT205 - DATA STRUCTURES

II SEMESTER

Unit I K1 Level

1. What is/are the following are as linear data structures?
   a) arrays b) linked list c) stack **d) all of these**
2. An array is a _____collection of _____ elements stored in adjacent memory locations.
   a) finite, similar b) similar, finite c) finite, non similar d) non similar, finite
3. Which function use to allocate the memory?
   a) calloc() **b) malloc()** c) realloc() d) all of these
4. The ---------- function is used to modifies a size of previously allocated space in memory
   a) Calloc () **b) realloc ()** c) malloc () d) allocate ()
5. In Linked List the **NULL** pointer is used to tell
   a) End of the List b) Beginning of the List c) Linked List is Empty d) None of these
6. Inserting an item into the stack when stack is not full is called _____ Operation and deletion of item form the stack, when stack is not empty is called _____operation.
   a) **push, pop** b) insert, delete c) pop, push d) delete, insert
7. Array is a --------
   a) **Linear data structure** b) Non-Linear data structure c) Data type d) List
8. NULL pointer is used to tell
   a) **end of linked list** b) empty pointer c) linked list is empty d) all of the above
9. The ---------- function is used to allocate a block of memory
   a) calloc() b) get() c) malloc() d) allocate()
10. The ---------- type of list have two way link.
    a) Circular linked list b) List c) Two way linked list **d) Doubly linked list**

Unit II

1. In which one made addition or deletion of elements takes place at the same end?
   a) Linked list b) array **c) stack** d) queue
2. What are the important functions involved during stack operations?
   a) push() b) pop **c) both (a) & (b)** d) none of the above.
3. Stack is……. Data structure
   a) Static b) **Dynamic** c) In built d) None of these
4. With the help of insert or delete an element in a stack is_____
   a) push b) pop **c) top** d) none of these
5. Stacks are called as _____

a) LIFO b) FIFO c) LILO d) FILO

6. If the operator symbols are placed before its operands, then the expression is in _____ notation

a) frontfix b. postfix **c. Infix** d. Prefix

7. _____ is very useful in situation when data have to stored and then retrieved in reverse order.

A Stack B Queue C List D Link list

8. The operation of processing each element in the list is known as

A Sorting B Merging C Inserting D **Traversal**

9. Which is/are the application(s) of stack

A) Function calls B) Large number Arithmetic
C) Evaluation of arithmetic expressions D) **All of the above**

10. Which of the following data structures are indexed structures?

A linear arrays B linked lists **C both of above** D none of above

Unit III

1. "FRONT=REAR" pointer refers to empty in ----------

a) Stack b) Array **c) Queue** d) Linked List

2. A data structure in which elements are added at one end and removed at another end is known as------

a) queue b) stack c) array d) none of these

3. Queues are called as _____

a) LIFO **b) FIFO** c) LILO d) FILO

4. A data structure in which insertion and deletion can take place at both the ends is called _____

a) deque b. stack c. Circular queue d. Priority queue

5. Which data structure allows deleting data elements from and inserting at rear?

A Stacks B Dequeues **C Queues** D Binary search tree

6. Identify the data structure which allows deletions at both ends of the list but insertion at only one end.

A) Stack B **Input restricted Dequeue** C Priority queues
D Output restricted Dequeue

7. A data structure where elements can be added or removed at either end but not in the middle

A Linked lists B Stacks C Queues D **Deque**

8. The situation when in a linked list START=NULL is

A **underflow** B overflow C housefull D saturated

9. ..... is not the operation that can be performed on queue.

a) Traversal b) insertion c) Deletion d) Retrieval

10. Which of the following is not the type of queue?

a. Priority queue b. Circular queue c. **Single ended queue** d. Ordinary queue

Unit IV

1. The method of search which traverse the data sequentially is called

   a) Binary search **b) Sequential search** c) Row major search d) Column major search

2. Maximum level of any node of a binary tree T is ----------

   a) Complete b) Binary **c) Height** d) Width

3. For a binary tree of height h the maximum number of nodes can be _____

   a) $2^{h-1}+1$ b) $2^{h-1}-1$ **c) $2^{h+1}-1$** d) $2^{h+1}+1$

4. Which searching method requires that all keys must reside in internal memory?

   a) Binary b) sequential c) Hasing d) None of these

5. Which one is not a traversing method?

   a) inorder b) preorder c) postorder **d) outorder**

6. The depth of a complete binary tree is given by

   A. Dn = n log2n B. Dn = n log2n+1 C. Dn = log2n D. **Dn = log2n+1**

7. _____ is a node, which has no sons.

   a) Father b. Ancestor **c. Leaf node** d. Siblings

8. Which of the following is non-liner data structure?

   A) Stacks B) List C) Strings D) **Trees**

9. The Node has no children is called -------- node.

   a) Root b) Siblings **c) Leaf** d) Father

10. ……. Is a directed tree in which outdegree of each node is less than or equal to two.

    A) Unary tree B) **Binary tree** C) Trinary tree D) Both B and C

Unit V

1. Quick sort was introduced in the year of ----------

   a) 1972 b) 1982 **c) 1962** d) 1952

2. Worst case Complexity of Binary Tree is ----------

   a) **0(n)** b) 0(2n) c) $0(2n^2)$ d) $0(n^3)$

3. In the sort, file is divided into sub files which are to be independently sorted and merged

   a) Quick sort b) **Heap Sort** c) Bubble sort d) None of these

4. What is time complexity of binary tree sort?

   a) **0(n)** b) 0(n log n) c) $0(n^2)$ d) none of these

5. A binary tree whose every node has either zero or two children is called

   a) **Complete binary tree** b) binary search tree
   c) extended binary tree d) none of these

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

   a) processor & memory b) **time & space** c) data & space d) complexity & capacity

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 complexity of Quick Sort is _____

a) O (n^2) b. O (n log n) c. O (n^3) d. O (log n)

9. The complexity of Binary search algorithm is

A) O(n) B) **O(log n)** C) O(n2) D) O(n log n)

10. The complexity of linear search algorithm is

A **O(n)** B O(log n) C O(n2) D O(n log n)

NGM COLLEGE (AUTONOMOUS) POLLACHI

DEPARTMENT OF INFORMATION TECHNOLOGY

18UIT205 - DATA STRUCTURES

II SEMESTER

K2 Level

Unit I

1. What is a data structure?
2. What does abstract data type means?
3. Explain about the types of linked lists
4. What are the disadvantages array implementations of linked list?
5. Whether Linked List is linear or Non-linear data structure?
6. What is the difference between a Linked list and an Array?
7. Define the term array. How are two-dimensional arrays represented in memory?
8. Explain how address of an element is calculated in a two dimensional array.
9. List out the areas in which data structures are applied extensively?
10. Let P be a pointer to a singly linked list. Show how this list may be used as a stack. That is, write algorithms to push and pop elements. Specify the value of P when the stack is empty

Unit II

1. Evaluate the following prefix expression " ++ 26 + - 1324".
2. Convert the following infix expression to post fix notation ((a+2)*(b+4)) -1.
3. Stack can be described as a pointer.
4. Give the statement to perform push operation.
5. Which data structure is needed to convert infix notations to post fix notations?
6. Parenthesis are never needed in prefix or postfix expressions. Why?
7. Explain how to implement two stacks in one array
8. Execute your algorithm to convert an infix expression to a post fix expression with the following infix expression as input A+B-C/D*E*F*G/H
9. Give the statement to perform pop operation.
10. Demonstrate the top notation in stack.

Unit III

1. Infer the circular queue
2. In which data structure, elements can be added or removed at either end, but not in the middle?
3. Minimum number of queues needed to implement the priority queue?
4. Write functions for inserting and deleting at either end.

5. Refer Front & Rear.
6. Write down routines for inserting and deleting elements from a circular queue implemented using arrays.
7. Interpret Front = Rear
8. How do you know the Queue is empty?
9. Illustrate the Deque.
10. What is the other name double ended queue?

Unit IV

1. What is a Binary Search Tree (BST)?
2. What is the maximum total number of nodes in a tree that has N levels? Note that the root is level (zero).
3. How many different binary trees can be made from three nodes that contain the key values 1, 2 & 3?
4. List the outline of binary tree traversal.
5. Outline the important nodes in binary tree
6. Show the example of completed binary tree.
7. Summarize the tree models.
8. Interpret the root node in binary tree.
9. Infer the black rule.
10. Illustrate the red rule

Unit V

1. Write a Binary Search.
2. How many different binary search trees can be made from three nodes that contain the key values 1, 2 & 3?
3. Which sorting algorithm is easily adaptable to singly linked lists?
4. What do you mean by complexity of an algorithm?
5. Explain the meaning of worst case analysis
6. Best case analysis with an example.
7. What is the average number of comparisons in a sequential search?
8. The element being searched for is not found in an array of 100 elements. What is the average number of comparisons needed in a sequential search to determine that the element is not there, if the elements are completely unordered?
9. Define Hashing.
10. Which sorting algorithm is best if the list is already sorted? Why?

NGM COLLEGE (AUTONOMOUS) POLLACHI

DEPARTMENT OF INFORMATION TECHNOLOGY

18UIT205 - DATA STRUCTURES

II SEMESTER

K3 Level

Unit I

1. Write an algorithm to insert a node in the beginning of the linked list.
2. Write a complete programme in C to create a single linked list.
3. Write functions to do the following operations

   (i) Insert a new node at the end (ii) Delete the first node
4. Enumerate various operations possible on ordered lists and arrays. Write procedures to insert and delete an element in to array.
5. Analyze doubly linked list with algorithm.

Unit II

1. Distinguish Stack and Recursion.
2. Analyze with polish notation.
3. Inspect the reversing a list in stack
4. Simplify the operations of ADT in Stack.
5. How will you motive to important stack notations.

Unit III

1. How do you build to representation of a queue as a linked list? Explain
2. Develop the Circular queue.
3. What is a plan to create Deque? Explain.
4. Analyze the Stack & Queue.
5. Distinguish with Queue and Linked list.

Unit IV

1. Prove the hypothesis that “A tree having ‘m’ nodes has exactly (m–1) edges or branches”.
2. Construct the binary tree for the following sequence of nodes in preorder and inorder respectively.
   Preorder : G, B, Q, A, C, K, F, P, D, E, R, H
   Inorder: Q, B, K, C, F, A, G, P, E, D, H, R

3. What is a height balanced tree? Explain how the height is balanced after addition/deletion of nodes in it?
4. Let a binary tree 'T' be in memory. Write a procedure to delete all terminal nodes of the tree.
5. Consider the following eight numbers 50, 33, 44, 22, 77, 35, 60 and 40. Display the construction of the binary by inserting the above numbers in the given order.

Unit V

1. What is quick sort? Sort the following array using quick sort method.
   24 56 47 35 10 90 82 31

2. Create a heap with following list of keys:
   8, 20, 9, 4, 15, 10, 7, 22, 3, 12

3. The following values are to be stored in a hash table
   25, 42, 96, 101, 102, 162, 197

4. Describe how the values are hashed by using division method of hashing with a table size of 7.

5. Use chaining as the method of collision resolution.

NGM COLLEGE (AUTONOMOUS) POLLACHI
DEPARTMENT OF INFORMATION TECHNOLOGY
18UIT205 - DATA STRUCTURES
II SEMESTER

K4 & K5 LEVEL

Unit I

1. Write an algorithm to merge two sorted arrays into a third array. Do not sort the third array.
2. Write the programs for Linked List (Insertion and Deletion) operations
3. Write an algorithm to evaluate a postfix expression. Execute your algorithm using the following postfix expression as your input: a b + c d +*f.
4. Categorize any five basic linked list operations.
5. Analyze circular linked list with algorithm.

Unit II

1. How functioning to represent the stacks through linked list?
2. Evaluate the postfix expression. (K5)
3. Conversion of infix to prefix notation
4. Evaluate the prefix expression. (K5)
5. How translate to recursive procedure to non recursive procedure using stack?

Unit III

1. Explain input restricted deques.
2. What is a motive to set the priority queue? Explain.
3. Assess with applications of Queue.
4. Explain in detail of output restricted Deques.
5. Conclude that the real life usages of Queue.

Unit IV

1. Make a BST for the following sequence of numbers 45, 36, 76, 23, 89, 115, 98, 39, 41, 56, 69, 48 Traverse the tree in Preorder, Inorder and postorder.
2. Two Binary Trees are similar if they are both empty or if they are both nonempty and left and right sub trees are similar. Write an algorithm to determine if two Binary Trees are similar.
3. Construct a binary tree whose nodes in inorder and preorder are given as follows:

Inorder: 10, 15, 17, 18, 20, 25, 30, 35, 38, 40, 50
Preorder: 20, 15, 10, 18, 17, 30, 25, 40, 35, 38, 50

4. Given the following inorder and preorder traversal reconstruct a binary tree Inorder sequence D, G, B, H, E, A, F, I, C Preorder sequence A, B, D, G, E, H, C, F, I
5. Make a BST for the following sequence of numbers. 45,32,90,34,68,72,15,24,30,66,11,50,10 Traverse the BST created in Preorder, Inorder and Postorder.
6. What is a Binary Tree? What is the maximum number of nodes possible in a Binary Tree of depth d. Explain the following terms with respect to Binary trees (i) Strictly Binary Tree (ii) Complete Binary Tree
7. What are B-trees? Construct a B-Tree of order 3 for the following set of

Input data: 69, 19, 43, 16, 25, 40, 132, 100, 145, 7, 15, 18.

Unit V

1. Write an algorithm to sort a given list using Quick sort method. Describe the behaviour of Quick sort when input is already sorted.
2. Sort the following list using Heap Sort

66, 33, 40, 20, 50, 88, 60, 11, 77, 30, 45, 65.

3. Draw the 11 item hash table resulting from hashing the keys: 12, 44, 13, 88, 23, 94, 11, 39, 20, 16 and 5 using the hash function h(i) = (2i+5) mod 11.
4. Sort the following list using Heap Sort technique, displaying each step.
   20, 12, 25 6, 10, 15, 13
5. Show the result of inserting the keys.
   F, S, Q, K, C, L, H, T, V, W, M, R, N , P, A, B, X, Y, D, Z, E in the order to an empty B-tree of degree-3.
6. What do you mean by hash clash? Explain in detail any one method to resolve hash collisions.