**Java Cipher Assignment -2**

Submitted by : Pranjal Ghate

**Q.1** What do you mean by a Data structure?

**Ans**. A data structure is a particular way of organizing data in a computer so that it can be used effectively. For example, we can store a list of items having the same data-type using the array data structure.

**Q.2** What are some of the applications of DS?

**Ans.** Queue - You stand in a queue . First come first server or First In first Out (FIFO).

Stack: You take a dish from the pile of dishes in restaurant. The top dish which you take was the last entered when some one washed the dished and made the stack. List In First Out (LIFO)

Searching: You search a word from dictionary where words are alphabetically sorted. You unknowingly use binary search. How? For example lets assume you are going to search the word “Queen”. You randomly open a page in the middle and the page is starting with word “K”. So you will go for next pages to search the word “Queen” as Q comes after K. Then you open another page from K page. Lets say the word start from T. So you know “Queen” will be before that but ofcourse after K. That is how you search the word Queen and is called binary search.

List - You family tree is a list structured as tree. If you dont have a children yet, you are the leaf node.

Graph: You go to school or office every day. there are 5 roads that takes you to your school/office from home but you prefer to choose the shortest distance. How? Data structure .

**Q.3** What are the advantages of a Linked list over an array?

**Ans**. The Size of LinkedList is incremented at runtime. But in Array it is not possible. Also the number of elements in the linked list are limited to the available memory space while the number of elements in the array is limited to the size of an array.LinkedList stores the elements dynamically while array stores statically.

**Q4**. Write the syntax in C to create a node in the singly linked list.

**Ans.** struct node

{

    int data;

    struct node \*next;

};

struct node \*head, \*ptr;

ptr = (struct node \*)malloc(sizeof(struct node));

**Q.5** What is the use of a doubly-linked list when compared to that of a singly

linked list?

**Ans.** Doubly linkedlist is complex type of list. In this node contains the pointer on next node as well as previous node.

**Q.6** What is the difference between an Array and Stack?

**Ans**. Array is a linear data structure in which insertion and deletion can take place in any position. The elements can be retrieved randomly in Arrays.

Stack is also a linear data structure in which insertion and deletion will take place at the top position alone. It follows LIFO order . The element in the top most position alone can be retrieved.

**Q.7** What are the minimum number of Queues needed to implement the priority

queue?

**Ans** .Two Queues are needed. One stores data elements and another stores priorities.

**Q8.** What are the different types of traversal techniques in a tree?

**Ans** .There are three type of traversal technique of tree.

1. Pre Order
2. In Order
3. Post Order

**Q9.** Why it is said that searching a node in a binary search tree is efficient than that of

a simple binary tree?

**Ans** .Binary Search tree is a data Structure which uses the concept of BST. It is having special quality that every node has at most two child and the left child of every node has value than node and right child has more value than node. Using this rule in BST is easy.

**Q10**. What are the applications of Graph DS?

**Ans .** Graphs are used in circuit networks where points of connection are drawn as vertices and component wires become the edges of the graph.It is also used in transport networks where stations are drawn as vertices and routes become the edges of the graph.

Graphs are helpful in drawing maps that draw regions as vertices and as edges.

**Q.11**. Can we apply Binary search algorithm to a sorted Linked list?

**Ans.** Yes Binary Search is applied on Sorted linked list, If the list is sorted and the count of sorted list is known.

**Q.12**. When can you tell that a Memory Leak will occur?

**Ans.** GNU malloc. Under Linux using GNU libc, the kernel and/or C run-time will sometimes detect memory allocation or usage errors without doing anything special in your code or using any external tool.

**Q.13** How will you check if a given Binary Tree is a Binary Search Tree or not?

**Ans** .Binary Search Tree always have two child. Where the condition is that the right child should always be greater then the left one.

**Q.14** Which data structure is ideal to perform recursion operation and why?

**Ans** .Stack. Because of its LIFO property it remembers its 'caller' so knows whom to return when the function has to return. Recursion. makes use of system stack for storing the return addresses of the function calls.

**Q.15**. What are some of the most important applications of a Stack?**Ans** .The stack can be used to convert some infix expression into its postfix equivalent, or prefix equivalent. These postfix or prefix notations are used in computers to express some expressions

**Q.16** Convert the below given expression to its equivalent Prefix And Postfix notations.

**Ans.** *🡨--- Question is Incomplete. Expression missing --🡪.*

**Q.17**.Sorting a stack using a temporary stack

**Ans**.Given below is the algo to sort a stack using a temporary stack :

1. Create a temporary stack say tmpStack.
2. While input stack is NOT empty do this:
   * Pop an element from input stack call it temp
   * while temporary stack is NOT empty and top of temporary stack is greater than temp,  
     pop from temporary stack and push it to the input stack
   * push temp in temporary stack
3. The sorted numbers are in tmpStack.

**Q.18.** Program to reverse a queue

**Ans** .Procedure to reverse a queue

1. Pop the elements from the queue and insert into the stack. (Topmost element of the stack is the last element of the queue)
2. Pop the elements of the stack to insert back into the queue. (The last element is the first one to be inserted into the queue)

**Q.19**. Program to reverse first k elements of a queue

**Ans.**

* Create an empty stack.
* Dequeue items from given queue and push the dequeued items to stack.
* Enqueue the contents of stack at the back of the queue
* Dequeue (size-k) elements from the front and enqueue them one by one to the same queue.

**Q.20** Program to return the nth node from the end in a linked list

**Ans.** Calculate the length of Linked List. Let the length be “L”.  
After that Print (L – n + 1)th node from the beginning of the Linked List.

**Q.21**. Reverse a linked list

**Ans .**

1.Initialize three pointers prev as NULL, curr as head and next as NULL.

2.Iterate trough the linked list. In loop, do following.  
 Before changing next of current,  
store next node  
next = curr->next

Now change next of current  
This is where actual reversing happens

curr->next = prev

Move prev and curr one step forward  
prev = curr  
curr = next.

**Q. 22**.Replace each element of the array by its rank in the array

**Ans** . The problem can be solved using maps. The array element and its index are stored in two columns in the map and later the index is replaced by the element's rank.

* Elements in a map will be stored in sorted order, so on iterating it, the elements will be in an increasing order sequence.
* Assign values for each element in increasing order starting from 1 to n, incrementing by 1 for each element.

**Q. 23**.Check if a given graph is a tree or not

**Ans**. *🡨--Question Incomplete, Graph is missing -->*

**Q.24.** Find out the Kth smallest element in an unsorted array

**Ans**. A simple solution is to sort the given array using a O(N log N) sorting algorithm like Merge sort, Heap sort etc and return the element at index k-1 in the sorted array. The time complexity of this solution will be O(N Log N).

**Q.25**. How to find the shortest path between two vertices?

**Ans .**

1.We start the recursive function with the starting vertex.

2.For each node. Whenever we visited one vertex we mark it and go for all its unvisited adjacent nodes. If we found the destination vertex then count the number else we go for recursive call.