**Java Assignment 2 by Abhishek Patel**

1. Data Structure is a way of collecting and organising data in such a way that we can perform operations on these data in an effective way. Data Structures is about rendering data elements in terms of some relationship, for better organization and storage.
2. **a. Arrays:** Implementation of other data structures, Execution of matrices and vectors, Dynamic memory allocation, Pointer container, Control tables.

b. **Stack:** Evaluation of expressions, Backtracking, Runtime memory management, Arrangement of books in a library.

**c. Queue:** Here, the data sent need not be received at the same rate at which it was sent. A certain system resource is to be shared between different processes.

**d. Linked-List:** Representation of sparse matrices, Non-contiguous data storage, Implementation of non-binary tree or other data structures, Dynamic memory management, Equalizing parenthesis, Symbol tables.

**e. Set:** Mapping of data, Common data storage.

**f. Graph:** Computer networking, Problem solutions involving 'Depth-First' search or 'Breadth-First' search algorithms, Representation of matrices, Study of molecular interactions in Chemistry.

**g. Tree:** Representation of data lists, Quickly accessible data storage, Representation of hierarchal data, Routing of algorithms.

**h. Hash-table:** Unique data representation, Implementation of caches, Array association, Locating entries in a table, Representation of objects, Database indexing.

**i. File:** Implementation of computer programs, Data comparison, Storage of data having varying data types.

1. a. It can grow ,shrink it means it has variable size, while size of array is fixed.

b. Insertion/deletion of an element at beginning in a linked list is O(1) operation while in array it is O(n).

c. Insertion/ deletion at end can be O(1) if we use rear pointer (like in queue).

1. struct Node\* new\_node = (struct Node\*) malloc(sizeof(struct Node));
2. A DLL can be traversed in both forward and backward direction.  
    The delete operation in DLL is more efficient if pointer to the node to be deleted is given.

In singly linked list, to delete a node, pointer to the previous node is needed. To get this previous node, sometimes the list is traversed. In DLL, we can get the previous node using previous pointer.

| **STACKS** | **ARRAY** |
| --- | --- |
| Stacks are based on the LIFO principle, i.e., the element inserted at the last, is the first element to come out of the list. | In the array the elements belong to indexes, i.e., if you want to get into the fourth element you have to write the variable name with its index or location within the square bracket eg arr[4] |
| Insertion and deletion in stacks takes place only from one end of the list called the top. | Insertion and deletion in array can be done at any index in the array. |
| Stack has a dynamic size. | Array has a fixed size. |
| Stack can contain elements of different data type. | Array contains elements of same data type. |
| We can do only linear search | We can do both linear and Binary search |

1. Two queue will be required to make a priority queue. One queue is used for actual storing of data and another for storing priorities.
2. Ways of traversing BST are:
   1. **Inorder**  
      In case of binary search trees (BST), Inorder traversal gives nodes in non-decreasing order. To get nodes of BST in non-increasing order, a variation of Inorder traversal where Inorder traversal s reversed can be used.
   2. **Preorder**  
      Preorder traversal is used to create a copy of the tree. Preorder traversal is also used to get prefix expression on of an expression tree.
   3. **Postorder**  
      Postorder traversal is used to delete the tree. Postorder traversal is also useful to get the postfix expression of an expression tree..
   4. **Level Order Traversal**

This is a different traversal than what we have covered above. Level order traversal follows BFS(Breadth-First Search) to visit/modify every node of the tree.

1. **Facebook:** Each user is represented as a vertex and two people are friends when there is an edge between two vertices. Similarly friend suggestion also uses graph theory concept.

**Google Maps:** Various locations are represented as vertices and the roads are represented as edges and graph theory is used to find shortest path between two nodes.

**Recommendations on e-commerce websites:**The “Recommendations for you” section on various e-commerce websites uses graph theory to recommend items of similar type to user’s choice.

Graph theory is also used to study molecules in chemistry and physics

1. Yes, Binary search is possible on the linked list if the list is ordered and you know the count of elements in list. But while sorting the list, you can access a single element at a time through a pointer to that node i.e. either a previous node or next node. This increases the traversal steps per element in linked list just to find the middle element. This makes it slow and inefficient.
2. In computer science, a memory leak is a type of resource leak that occurs when a [computer program](https://en.wikipedia.org/wiki/Computer_program) incorrectly manages [memory allocations](https://en.wikipedia.org/wiki/Memory_allocation)in a way that memory which is no longer needed is not released. A memory leak may also happen when an [object](https://en.wikipedia.org/wiki/Object_(computer_science)) is stored in memory but cannot be accessed by the running code.
3. A [Binary Search Tree (BST)](https://www.educative.io/shoteditor/5038758772604928/preview) is a binary tree with the following properties:
   1. The left subtree of a particular node will always contain nodes whose keys are less than that node’s key.
   2. The right subtree of a particular node will always contain nodes with keys greater than that node’s key.
   3. The left and right subtree of a particular node will also, in turn, be binary search trees.
4. Stack. Because of its LIFO (Last In First Out) 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.

Every recursive function has its equivalent iterative (non-recursive) function.

1. Stacks can be used for expression evaluation.
   1. Stacks can be used to check parenthesis matching in an expression.
   2. Stacks can be used for Conversion from one form of expression to another.Stacks can be used for Memory Management.
   3. Stack data structures are used in backtracking problems.