WEEK ONE

DATA STRUCTURES AND ALGORITHMS

Data structures are basically just specialized formats for organizing and storing data

1. **Stacks**
   In a stack you can only insert or delete items at the top of the stack. The type of service in a stack is LIFO (Last In First Out). This means that the last item put in the stack is the first item to come out. The functions that can be used in a stack are:
   - Push – It is used to place data in to the stack
   - Pop – It is used to remove the top element of a stack
   - Peek – It is used to display the top element of a stack
   - Length/size – It is used to determine the number of elements in a stack

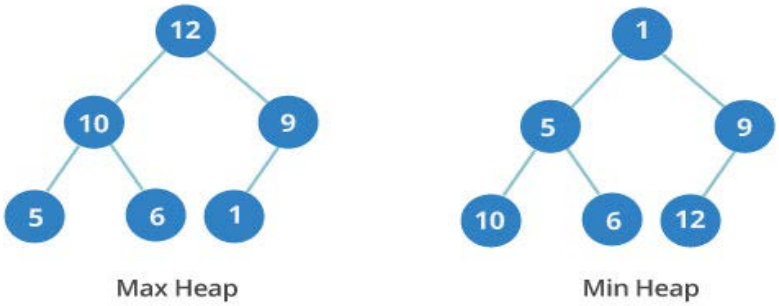
   A good example of a stack can be the browser's back button

2. **Queues**
   A queue is a way to hold data. A queue is similar to a stack but the type of service in a stack is LIFO while in a queue it is FIFO meaning First In First Out. Once a new element is added, all elements that were added before it has to be removed before the new element can be removed. A queue has two main operations:
   - Enqueue – this means to insert an element into the back of the queue
   - Dequeue – this means removing the front item
3. **Sets**
   Sets are like an array but does not store repeated items and the values are in no particular order. A set can be used to check the presence of an item. The add and remove functions can be used to add and remove elements in a set. The other important functions are:
   - Union – It combines all the items in two different sets and returns this as a new set. The new set has no duplicates.
   - Intersection – When given two sets it returns a new set with elements that are in both sets.
   - Difference – It returns a list of items that are in one set and not the other
   - Subset – It returns a Boolean value that shows if all the elements in a set are included in a different set.
4. **Trie**
   It is used to store associative data structures. A trie stores data in steps where each step is a node in the trie. Each node represents one letter of a word. You follow the branches of a trie in order to spell one word. Tries are used to store words for quick lookup, such as a word auto-complete feature
5. **Linked Lists**
   A linked list consists of a group of nodes which together represent a sequence. Each node contains two things: the data being stored and a link pointing to the next node in the sequence. There are also doubly linked lists where each node has two links pointing to both the next item and the previous item.
   The most basic operations in linked lists are adding an item to the list, removing an item from the list and searching for an item in the list.
6. **Binary Search Tree**
   It is used to hold data that when visualized, it looks like a tree. They allow fast lookup, addition and removal of items. A binary search tree has two branches for every node.

Binary search trees are ordered that is each left subtree is less than or equal to the parent node and each right subtree is greater than or equal to the parent node.

7. **Heap**

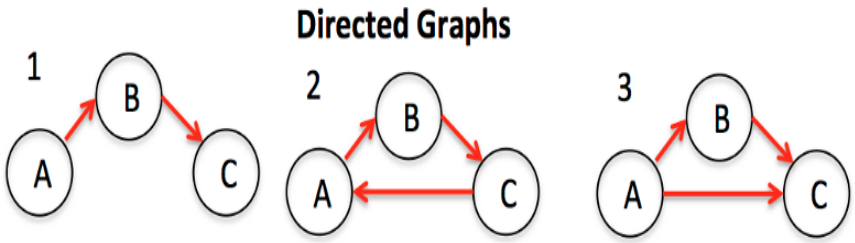
A heap is another type of data structure. Each node has at most two nodes. It is also a complete tree meaning all levels are fully filled. A heap can either be a max heap or a min heap. In max heap, the keys of the parent node are greater than or equal to the child node while in a min heap the keys of a child node are greater than or equal to the parent nodes. The main commands in a heap are inserting and removing.



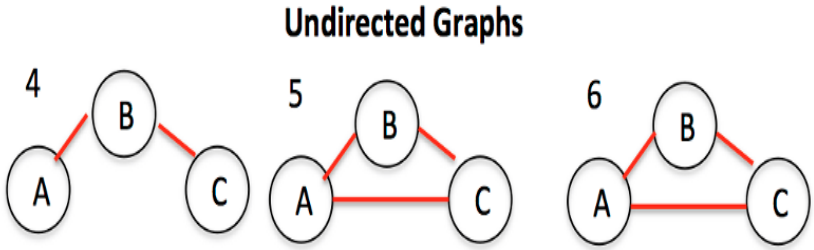
8. **Graph**

Graphs are collection of nodes and the connections between them. Data in a graph is called nodes or vertices while the connection between the nodes is called edges. An example of a graph is a social network where the nodes are people and the edges are friendships. There are two types of graphs:

- Directed graphs
  A directed graph has direction on its edges

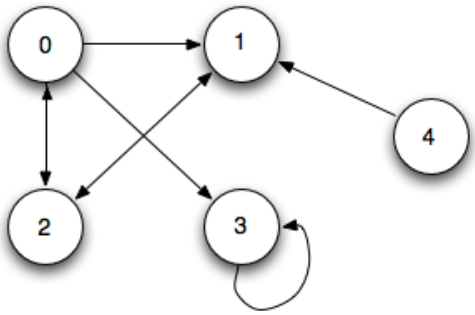


- Undirected graphs
  In undirected graphs there are no directions on the edges between nodes.



WAYS OF REPRESENTING A MATRIX

- **Adjacency Matrix**
  It is a grid of numbers where each row or column represents a different node in the graph. At the intersection of a row and a column is a number that indicates the relationship. Zero means there is no edge or relationship. Ones show that there is a relationship



	0	1	2	3	4
0	0	1	1	1	0
1	0	0	1	0	0
2	1	1	0	0	0
3	0	0	0	1	0
4	0	1	0	0	0

- **Adjacency List**
  This can be represented as a list where the left side is the node and the right-side lists all the other nodes it is connected
- **Incidence Matrix**
  Unlike adjacency matrix where both rows and columns represent nodes, in incidence matrix rows represent nodes while columns represent edges.

Traversal algorithms are algorithms used to visit a node in a graph. The main types of traversal algorithms are:

I. Breadth-first search – It is used to determine how far away a node is from the root node.
II. Depth-first search