Module 0

**

Data structure

- Specialized format for organizing and storing data in memory so that it can be used efficiently and easily
- Operations are insertion, deletion, traversal, searching, sorting
- Notations
- Infix (operators written in between operands)
- Postfix (operators are written after their operands
- Prefix (operators are written before their operands

Linear data structures

- Homogeneous elements
- Elements are in a sequence and form a linear series
- Easy to implement, since computer memory is linearly organized
- Ex. stack, queue, linked lists
- Queues and stacks can be implemented using arrays and linked lists

Non-linear data structures

- Data item is connected to several other data items
- Exhibit either hierarchical or parent-child relationship
- Not sequentially arranged
- Ex. trees, graphs

Stack

- Ordered list
- LIFO or FILO
- Insert with push()
- Delete with pop()
- Point to current newest element with top()
- Insertion and removal only allowed at one end: top
- Stack overflow is a full stack
- Stack underflow is an empty stack
- Ex. undo mechanisms, recursion/function calling, expression conversion/evaluation, reversing words

- Can be implemented with 2 queues
- Used for DFS
- DFS can also be implemented using recursion (recursion also uses function call stack)

Queue

- Ordered list
- FIFO
- Inserted at rear end aka enqueue
- Removed from front end aka dequeue
- Ex. hardware queues, network routers, operating systems (CPU scheduling), airport takeoff
- Can be implemented with 2 stacks
- Used for BFS

Priority queue

- Each item has some predefined priority
- Does not support FIFO
- An element can be inserted or removed from any position depending on their priority
- insert() adding item at array end in O(1) time
- getHighestPriority() linearly searches the highest priority item in array in O(n) time
- deleteHighestPriority() first linearly searches an item, then removes the item by moving all subsequent items one position back
- Using linked list, time complexity of all operations remains same as array
- It can make deleteHighestPriority() more efficient as we don't have to move items

Disadvantage of arrays

- Static memory allocation
- Max size reserved in advance
- Problems of less resource utilization
- Different data types cannot be stored in an array

Linked list

- Group of nodes in a sequence
- Each node contains data and node references
- Ex. making any list, basic of dynamic memory management, efficient memory management
- Singly linked
- Node has data and reference to next node
- Operations are insertion, deletion, traversal

- Doubly linked
- Node has data and reference to next and previous node
- Circularly linked
- Can be singly or doubly linked
- First and last node are connected
- No null node at the end

Tree

- Binary
- Hierarchical
- Each node has 2 children at most (left and right child)
- Topmost node is the root
- Is BST if the inorder traversal of a binary tree is sorted
- Ex. searching, time, geography, ancestry

Graph

- A pictorial representation of a set of objects, where some pairs of objects are connected by links
- The interconnected objects are represented by points termed as vertices (V) and the links that connect the vertices are called edges (E)
- A pair of sets (V,E) where V is a set of vertices and E is a set of edges, connecting the pairs of vertices
- Ex. almost any complex problem, network nodes
- Undirected
- Loop
- Multiple arc
- Directed
- Weighted

Recursion

- A procedure calls itself
- Procedure a calls procedure b which in turn calls procedure a again
- Visualizing
- Each node can be a procedure
- Each edge (connection) represents a procedure call
- Forms a circle
- Recursion first builds a stack with recursive calls
- It then pops the results (return values) in the opposite direction

**