

مالي دخل اذا جاكم شيء منها وطلع غلط ، لأن الموضوع ذا مشكلة

//mostafa massoud

## Linked List

Linked list is a data structure which links each node to the next node. The developer can use linked list in the following use cases.

- When the developer needs constant time for insertion and deletion.
- When the data dynamically grows.
- Do not access random elements from the linked list.
- Insert the element in any position of the list.

## Doubly Linked List

Doubly linked is a data structure which each node contain data and two links. One link point to previous node and another link point to next node. The developer can use doubly linked list in the following uses cases.

- Easier to delete the node from doubly linked list.
- Can be iterated in reverse order without recursion implementation.
- Insert or remove from double-linked lists faster.

## Stack

The stack is a last in, first out data structure. The developer can use the stack in the following use cases.

- Expression evaluation and syntax parsing.
- Finding the correct path in a maze using backtracking.
- Runtime memory management.
- Recursion function.

## Queue

The queue is a first in, first out (FIFO) data structure. The developer can use Queue in the following use cases.

- Use queue when the developer wants an order.
- Processed in First In First Out order.
- If the developer wants to add or remove both ends, they can use the queue or a double-ended queue.

## Binary Tree

A binary tree is a tree data structure in which each node has at most two child nodes. The developer can use Binary Tree in the following use cases.

- Find name in the phone book.
- Sorted traversal of the tree.
- Find the next closest element.
- Find all elements less than or greater than a certain value.

# Binary Search Tree

A binary search tree is a tree data structure in which root node is less than left subtree and greater than right subtree. The developer can use Binary Search Tree in the following use cases.

- Binary Search Trees are memory-efficient.
- Use when the data need to be sorted.
- Search can be done for a range of values.
- Height balancing helps to reduce the running time.
- 
- **Applications of Priority Queue:**
  - 1) CPU Scheduling
  - 2) Graph algorithms like [Dijkstra's shortest path algorithm](#), [Prim's Minimum Spanning Tree](#), etc
  - 3) All [queue applications](#) where priority is involved.
- 

# Heap

A heap is a specialized tree-based abstract data type that satisfies the heap property. The developer can use Heap in the following use cases.

- Implement Priority Queue.
- whenever the developer want quick access to the largest (or smallest) item.
- Good for selection algorithms (finding the min or max).
- Operations tend to be faster than for a binary tree.
- Heap sort sorting methods being in-place and with no quadratic worst-case scenarios.
- Graph algorithms are using heaps as internal traversal data structures, run time will be reduced by polynomial order.

# Hashing

Hash table is a data structure used to implement an associative array, a structure that can map keys to values. The developer can use Hash table in the following use cases.

- Constant time operation.
- Inserts are generally slow, reads are faster than trees.
- Hashing is used so that searching a database can be done more efficiently.
- Internet routers use hash tables to route the data from one computer to another.
- Internet search engine uses hash function effectively.

# Graph

The graph is an abstract data type that is meant to implement the graph and directed graph concepts from mathematics. The developer can use Graph in the following use cases.

- Networks have many uses in the practical side of graph theory.

- Finding the shortest path between the cities.
- Solve maze game.
- Find the optimized route between the cities.

## Array

The array is a data structure to store the same type of elements continuously. The developer can use an array in the following use cases.

- Need access the elements using the index.
- Know the size of the array before defining the memory.
- Speed when iterating through all the elements in the sequence.
- Array takes less memory compare than linked list.

## B-Tree

B-tree is a tree data structure that keeps data sorted and allows searches, sequential access, insertions, and deletions in logarithmic time. The developer can use B-Tree in the following use cases.

- File systems.
- Database operations.
- .

## AVL Tree

AVL tree, the shape of the tree is constrained at all times such that the tree shape is balanced. The height of the tree never exceeds  $O(\log n)$ . The developer can use AVL Tree in the following use cases.

- When the developer wants to control the tree height outside -1 to 1 range.
- Fast looking element.
- AVL trees are beneficial in the cases where you are designing some database where insertions and deletions are not that frequent but you have to frequently look-up for the items present in there.
- AVL Trees seem to be the best data structure for Database Theory (Additionally, AVL Trees are used for all sorts of in-memory collections such as sets and dictionaries.