Data Structure operations and time complexity:

Arrays:

Set, Check element at a particular index: O(1)

Searching: O(n) if array is unsorted and O(log n) if array is sorted and something like a binary search is used,

As pointed out by Aivean, there is no Delete operation available on Arrays. We can symbolically delete an element by setting it to some specific value, e.g., -1, 0, etc. depending on our requirements

Similarly, insert for arrays is basically Set as mentioned in the beginning

-- ----------

Linked List:

Inserting: O (1), if done at the head, O(n) if anywhere else since we have to reach that position by

traversing the linked list linearly.

Deleting: O (1), if done at the head, O(n) if anywhere else since we have to reach that position by

traversing the linked list linearly.

Searching: O(n)

Stack:

Push**:** O(1)

Pop**:** O(1)

Top**:** O(1)

Search **(Something like lookup, as a special operation):** O(n)

-- -------------

Queue/Deque/Circular Queue:

Insert**:** O(1)

Remove**:** O(1)

Size**:** O(1)

-- -------------

Binary Search Tree:

Insert, delete and search**: Average case:** O(log n)**, Worst Case:** O(n)

HashMap/Hashtable/HashSet:

Insert/Delete**:** O(1) **amortized**

Re-size/hash**:** O(n)

Contains**:** O(1)

-- -------------

Heap/PriorityQueue (min/max):

Find Min/Find Max: O(1)

Insert: O(log n)

Delete Min/Delete Max: O(log n)

Extract Min/Extract Max: O(log n)

Lookup, Delete (if at all provided): O(n), we will have to scan all the elements as they are not ordered like BST

The data structure is a particular way of organizing data in a computer. The developer must choose the appropriate data structure for better performance. If the developer chooses a bad data structure, the system does not perform well. This article explains each data structure's advantages and usage.

Linked List

The linked list is a data structure that links each node to the next node. The developer can use the 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.

Circular Linked List

A circular linked list is a linked list in which the link field of the tail node link to the head node. The developer can use a circular linked list in the following use cases.

- Develop the buffer memory.
- Represent a deck of cards in a game.
- Browser cache allows hitting the BACK button.
- Implement the Most Recently Used (MRU) list.
- Undo functionality in Photoshop or Word.

Doubly Linked List

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

- Easier to delete the node from the doubly linked list.
- It 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 a 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 the 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 the root node is less than or equal to the left subtree and greater than or equal to the 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.

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 wants 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, the 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 a 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.
- The Internet search engine uses a hash functions 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 the maze game.
- Find the optimized route between the cities.

Red-Black Tree

Red–black tree is a binary search tree with an extra bit of data per node, its color, which can be either red or black. The developer can use Red-Black Tree in the following use cases.

- Java TreeMap and C++ map implemented using Red Block Tree.
- Computational Geometry Data structures.
- Scheduler applications.

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 to 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.
- The array takes less memory compare than a linked list.

Matrix

Matrix is a data structure that stores the data using rows and columns. The developer can use Matrix in the following use cases.

- Matrix arithmetic in graphic processing algorithms.
- Represent the graph.
- Represent quadratic forms and linear algebra solution.

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.

Splay Tree

A splay tree is a self-adjusting binary search tree with the additional property that recently accessed elements are quick to access again. The developer can use Splay Tree in the following use cases.

- When the developer wants to access to the recent data easily.
- Allow duplicate items.
- Simple implementation and take less memory.
- When the application deals with a lot of data, use the splay-tree.

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.

Trie

A Trie (digital tree and sometimes radix tree or prefix tree), is an ordered tree data structure that is used to store a dynamic set or associative array where the keys are usually strings. The developer can use Trie in the following use cases.

- Fixed dictionary and want to look up quickly.
- Require less storage for a large dictionary.
- Matching sentences during string matching.
- Predictable O(k) lookup time where k is the size of the key.
- Lookup can take less than k time if it's not there.
- Supports ordered traversal.
- No need for a hash function.
- Deletion is straightforward.

Minimum Spanning Tree

A spanning tree of that graph is a subgraph that is a tree and connects all the vertices together. A minimum spanning tree (MST) or minimum weight spanning tree is then a spanning tree with weight less than or equal to the weight of every other spanning tree. The developer can use Minimum Spanning Tree in the following use cases.

- Describe financial markets.
- Handwriting recognition of mathematical expressions.
- Image registration and segmentation.
- Constructing trees for broadcasting in computer networks.

We discussed different data structures and uses cases to choose the appropriate data structure. When the candidate attends the technical coding interview or uses the application programming interface in software development, the candidate must choose the correct data structure. If the candidate uses the incorrect data structure, it may work. But the programs may fail with more data or with the different use cases.
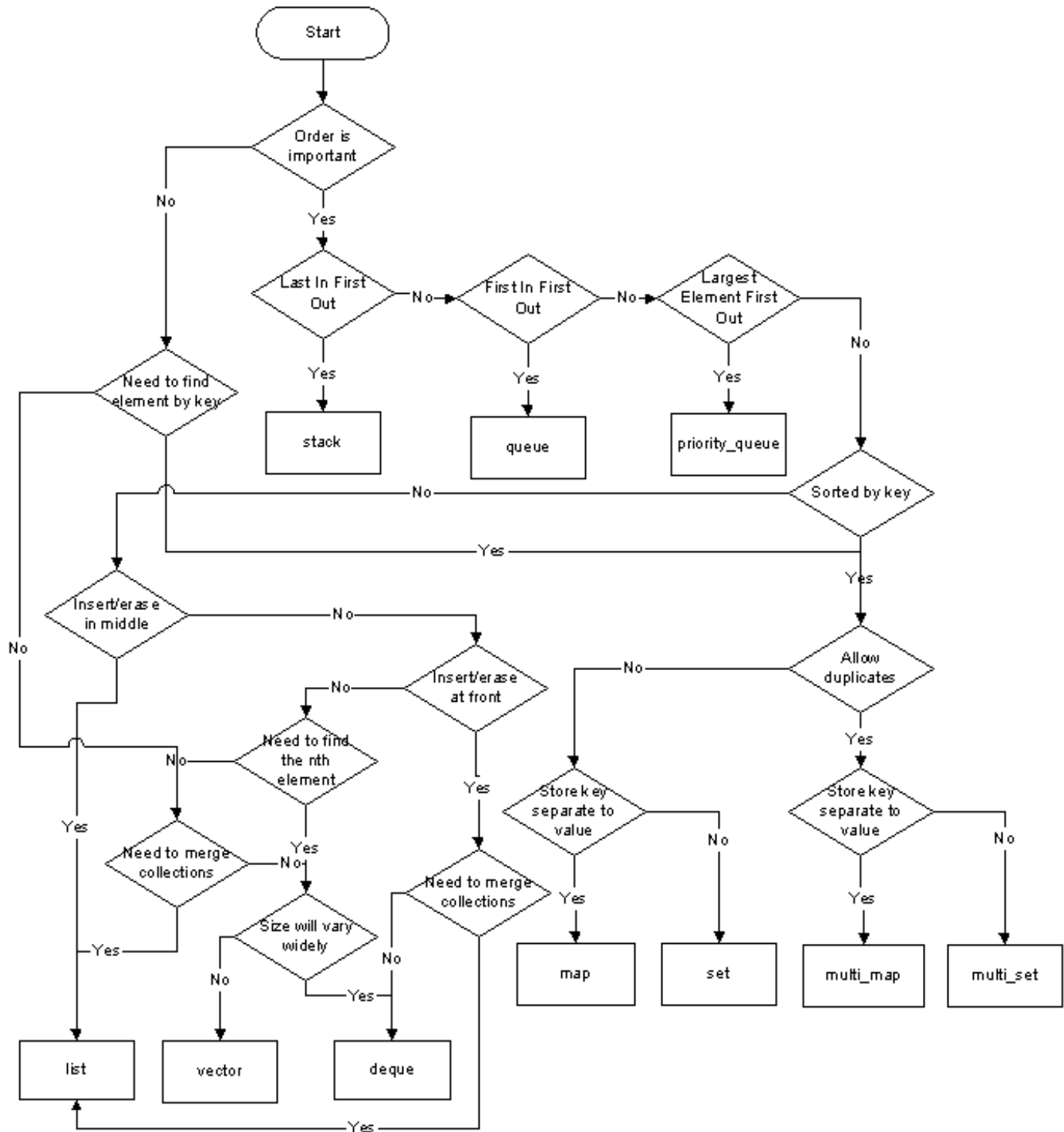
Real-time application of Data Structures

- Difficulty Level : Medium

A data structure is a particular way of organizing data in a computer so that it can be used effectively. In this article, the real-time applications of all the data structures are discussed.

Application of Arrays:

Arrays are the simplest data structures that stores items of the same data type. A basic application of Arrays can be storing data in tabular format. For example, if we wish to store the contacts on our phone, then the software will simply place all our contacts in an array.

Some other applications of the arrays are:

1. Arrangement of leader-board of a game can be done simply through arrays to store the score and arrange them in descending order to clearly make out the rank of each player in the game.
2. A simple question Paper is an array of numbered questions with each of them assigned to some marks.
3. 2D arrays, commonly known as, matrix, are used in image processing.
4. It is also used in speech processing, in which each speech signal is an array.

Application of Linked Lists:

A linked list is a sequence data structure, which connects elements, called nodes, through links.

Singly Linked list
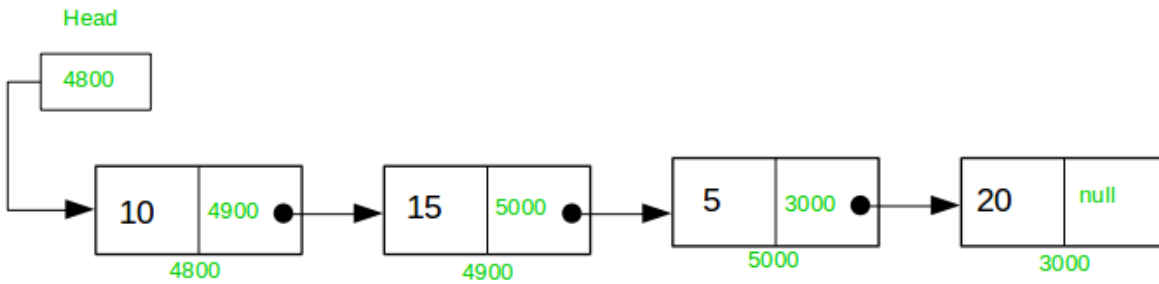
Some other applications of the linked list are:

1. Images are linked with each other. So, an image viewer software uses a linked list to view the previous and the next images using the previous and next buttons.
2. Web pages can be accessed using the previous and the next URL links which are linked using linked list.
3. The music players also use the same technique to switch between music.
4. To keep the track of turns in a multi player game, a circular linked list is used.

Application of Stack:

A stack is a data structure which uses LIFO order.
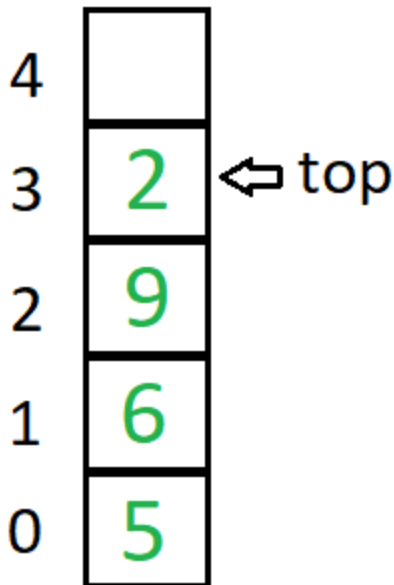
Stack

Some Applications of a stack are:

1. Converting infix to postfix expressions.
2. Undo operation is also carried out through stacks.
3. Syntaxes in languages are parsed using stacks.
4. It is used in many virtual machines like JVM.
5. Forward – backward surfing in browser
6. History of visited websites
7. Message logs and all messages you get are arranged in stack
8. Call logs, E-mails, Google photos' any gallery , YouTube downloads, Notifications (latest appears first)
9. Scratch card's earned after Google pay transaction

Application of Queue:

A queue is a data structure that uses FIFO order.
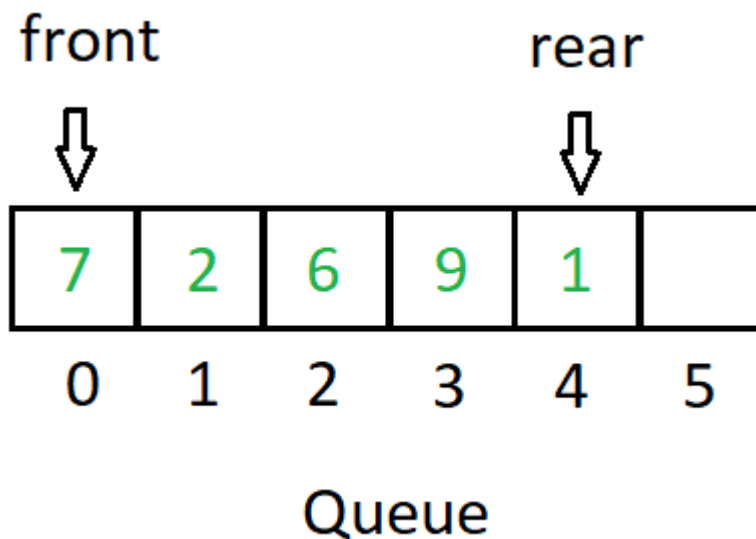
Some applications of a queue are:

1. Operating System uses queue for job scheduling.
2. To handle congestion in networking queue can be used.
3. Data packets in communication are arranged in queue format.
4. Sending an E-mail, it will be queued
5. server while responding request
6. Uploading and downloading photo's, first kept for uploading/downloading will completed first (Not if there is threading)
7. Most of internet requests and processes uses queue
8. While switching multiple applications, windows uses circular queue.

Application of Graph:

Graph is a data structure where data is stored in a collection of interconnected vertices (nodes) and edges (paths).
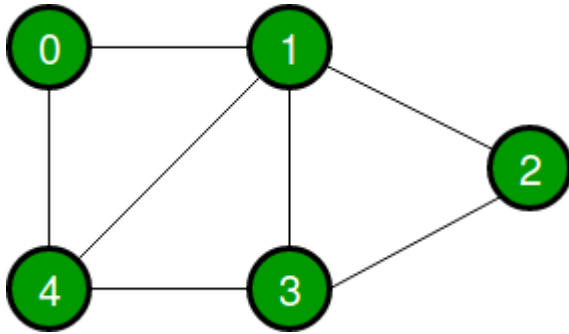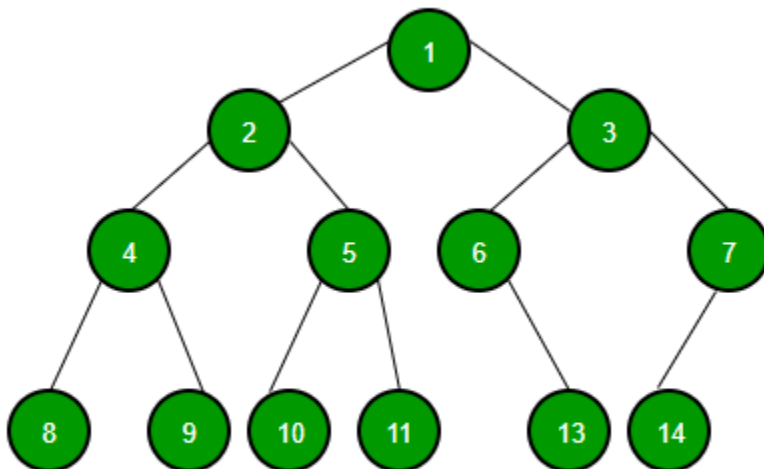
Some applications of a graph are:

1. Facebook's Graph API uses the structure of Graphs.
2. Google's Knowledge Graph also has to do something with Graph.
3. Dijkstra algorithm or the shortest path first algorithm also uses graph structure to finding the smallest path between the nodes of the graph.
4. GPS navigation system also uses shortest path APIs.
5. Networking components has huge application of graph
6. Facebook, Instagram and all social media networking sites every user is Node
7. Data organization

Application of Tree:

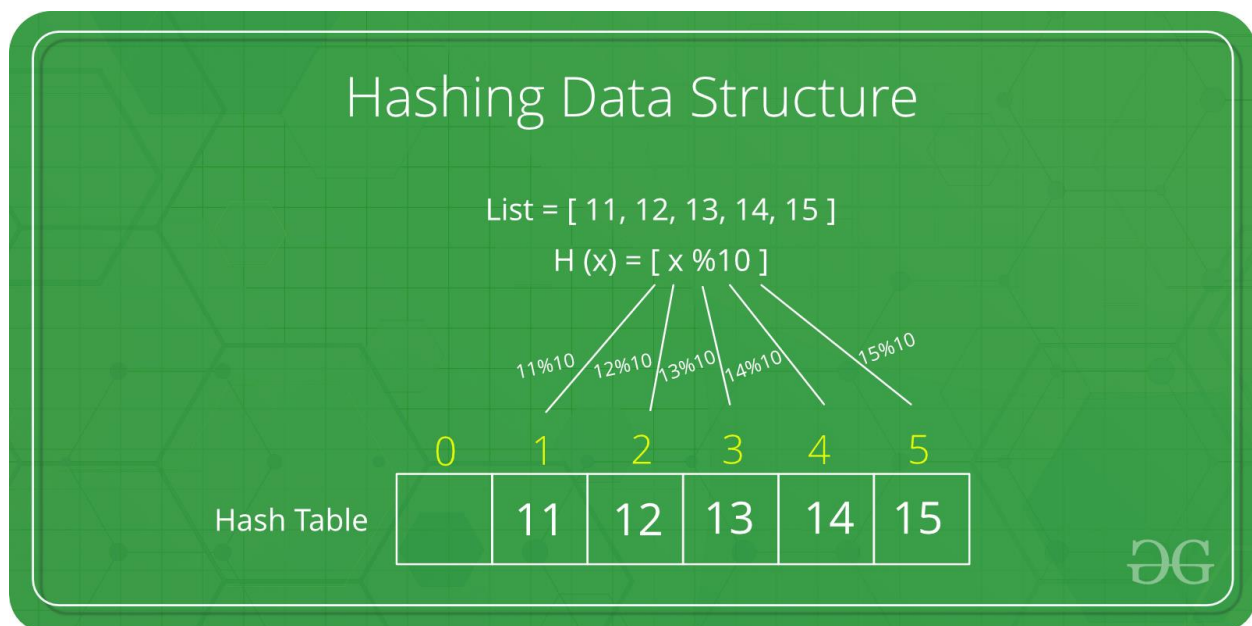Trees are hierarchical structures having a single root node.

Some applications of the trees are:

1. XML Parser uses tree algorithms.
2. Decision-based algorithm is used in machine learning which works upon the algorithm of tree.

3. Databases also uses tree data structures for indexing.
4. Domain Name Server(DNS) also uses tree structures.
5. File explorer/my computer of mobile/any computer
6. BST used in computer Graphics
7. Posting questions on websites like Quora, the comments are child of questions

[Application of Hash Tables](#)**:**
Hash Tables are store data in key-value pairs. It only stores data which has a key associated with it. Inserting and Searching operations are easily manageable while using Hash Tables.

Hashing Data Structure

List = [11, 12, 13, 14, 15]

$H(x) = [\ x\ \%10\ ]$

$11\%10$ $12\%10$ $13\%10$ $14\%10$ $15\%10$

| 0	1	2	3	4	5
	11	12	13	14	15

Hash Table

Some applications of a hashtable are:

1. Data stored in databases is generally of the key-value format which is done through hash tables.
2. Every time we type something to be searched in google chrome or other browsers, it generates the desired output based on the principle of hashing.
3. Message Digest, a function of cryptography also uses hashing for creating output in such a manner that reaching to the original input from that generated output is almost next to impossible.
4. In our computers we have various files stored in it, each file has two very crucial information that is, filename and file path, in order to make a connection between the filename to its corresponding file path hash tables are used.

[Application of Heap](#)**:**
A Heap is a special case of a binary tree where the parent nodes are compared to their children with their values and are arranged accordingly.

Heap Data Structure
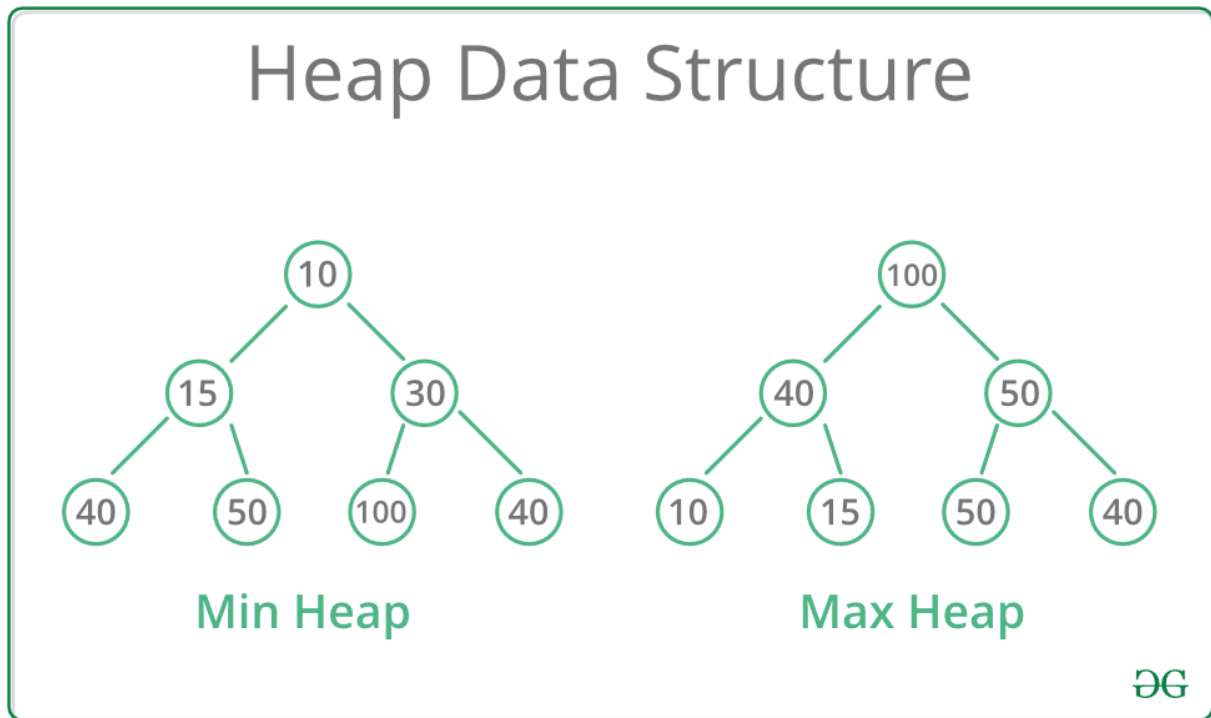
Some applications of heaps are:

1. In [heapsort Algorithm](#), which is an algorithm for sorting elements in either [min heap](#)(the key of the parent is less than or equal to those of its children) or [max heap](#)(the key of the parent is greater than or equal to those of its children), [sorting](#) is done with the creation of heaps.
2. Heaps are used to implement a [priority queue](#) where priority is based on the order of heap created.
3. Systems concerned with security and embedded system such as Linux Kernel uses Heap Sort because of the O(n log(n)).
4. If we are stuck in finding the [Kthsmallest (or largest) value of a number](#) then heaps can solve the problem in an easy and fast manner.

[Application of Matrix](#):
Matrix is an ordered collection of columns and rows of elements. It is necessary to enclose the elements of a matrix within the brackets.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Some applications of a matrix are:

1. In geology, matrices are used for making seismic surveys.
2. Used for plotting graphs, statistics and also to do scientific studies and research in almost different fields.
3. Matrices are also used in representing the real-world data's like the population of people, infant mortality rate, etc.
4. They are best representation methods for plotting surveys.