Data Structures:

Linked Lists:

- Definition: Linear data structure where each element is a separate object
- Each node has a head (reference) and the datan | Data | Next | ->
- Single-linked list, doubly-linked list, and circular linked list
- Methods: Add-first, Traverse, Add-last, Insert after, Insert before, Delete, Iterate, Clone

Trees, Tries, & Graphs:

- Graph is a data structure that consists of: a finite set of vertices also called nodes, and a finite set of ordered pairs of the form (u,v) called an edge, not the same as (v, u) where v is the number of vertices and e is the number of edges
- Graphs are classified on the basis of: Direction (bidirectional or unidirectional) and weight (Weight is associated with edges)
- Trie is an efficient data structure for searching words in dictionaries
- Search complexity with trie is linear in terms of word or key length to be searched
- No collisions in a trie and much faster that BST O(max string length)
- Tries require a lot of extra space, common example is dictionaries due to prefix search capability
- Segment tree is usually implemented when there are a lot of queries on a set of values, involving max, min, and sum on an input range of given set.
- Implemented using an array, used when we need to find max/min/sum/product of numbers in a range

Stacks & Queues:

- A stack is a container of objects that are inserted and removed according to the last-in first-out principle. Push the item into the stack and pop the item out of the stack.
- Stack is a limited access data structures, elements can only be added and removed from the stack only at the top, recursive data structure
- Simplest application is to reverse a word
- A queue is a container of objects (a linear collection) that is inserted and removed according to the first in first out principle. Excellent example is a line of students in the food court.
- Enqueue (add to back) and dequeue (remove from front)

Heaps:

- Specialized tree-based data structure which is essentially an almost complete tree that satisfies the heap property
- Highest or lowest priority element is always stored at the root, can be regarded as partially ordered

Vectors/ArrayLists:

- Vectors are the same as dynamic arrays with the ability to resize itself automatically when an element is inserted or deleted, with their storage being handled automatically by the container
- Vector elements are placed in contiguous storage so that they can be accessed and traversed using iterators. Data is inserted at the end of the vector
- Inserting at the end of a vector takes differential time, as sometimes there may be a need of extending the array
- Removing the last element takes only constant time because no resizing happens
- Inserting and erasing at the beginning or in the middle is linear in time
- ArrayList is a part of collection framework and is present in java.util package. Provides dynamic arrays in Java. May be slower than standard arrays but can be helpful in programs where lots of manipulation in the array is needed

Hash Tables:

- Performs extremely well compared to data structures like Array, LinkedList, Balanced BST, etc. O(1) search time on average and O(n) in worst case
- Function that converts a big number to a small practical integer value, which is used as an index in hash table. Essentially, it maps a big number or string to a small integer that can be used as index in a hash table
- Good hash function should be efficiently computable and should uniformly distribute the keys
- Hash table is an array that stores pointers to records corresponding to a given phone number. Null if no existing number has a hash function value equal to the index for the entry
- Collision handling used for two key results in the same value.
- Chaining: Make each cell of hash table point to a linked list of records that have the same hash function value. Requires additional memory outside the table
- Open addressing: All elements are stored in the hash table itself. Examine each slot until the desired element is found or it is clear that the element is not in the table.

Algorithms:

Breadth-First Search:

- Algorithm for traversing or searching tree or graph data structures
- Starts at the tree root (or some arbitrary node of a graph, sometimes referred to as a ‘search key’ and explores all of the neighbor nodes at the present depth prior to moving on to the nodes at the next depth level

- Accomplished by enqueueing each level of a tree sequentially as the root of any subtree is encountered. Two cases in iterative algorithm: Root case and general case
- Root case: traversal queue is initially empty so the root node must be added before the general case
- General case: Process any items in the queue, while also expanding their children. Stop if queue is empty. General case will halt after processing the bottom level as leaf nodes have no children

Depth-First Search:

- Algorithm for traversing or searching tree or graph data structures
- Algorithm starts at the root node (selecting some arbitrary node as the root node in the case of a graph) and explores as far as possible along each branch before backtracking

Binary Search:

- Idea of a binary search is to use the information that the array is sorted and reduce the time complexity to O(Log n)
- Basically ignore half of the elements after one comparison
- Compare x with the middle element
- If x matches middle, return middle index
- If x is greater than the middle element, the x can only lie in the right half subarray after the mid element. So, recur to the right half.
- Else, recur to the left half.

Merge Sort:

- Divide and Conquer algorithm. Divides input array in two halves, calls itself for the two halves and then merges the two sorted halves. Merge() function is used for merging two halves. Merge is the key process that assumes that the arrays are sorted and merges the two sorted sub-array into one
- Breaks array down into each individual part, then resorts all of the previous parts back together

Quick Sort:

- Divide and Conquer algorithm. Picks an element as pivot and partitions the given array around the picked pivot. There are many different versions of quickSort that pick pivot in different ways: First element as pivot, last element as pivot, random element as pivot, median as pivot.
- Key process in quickSort() is partition. Given an array and an element x of array as pivot, put x at its correct position in sorted array and put all smaller elements (smaller than x)

before x, and put all greater elements (greater than x) after x. Should be done in O(n) time

Concepts:

Bit Manipulation:

- Process of applying logical operations on a sequence of bits to achieve a required result
- Binary operations with 0 and 1: AND, OR, XOR, NOT, RIGHT SHIFT, LEFT SHIFT

Memory (Stack v. Heap):

- Stack is a special region of your computer's memory that stores temporary variables created by each function. Uses a LIFO data structure that is managed and optimized by the CPU quite closely. Every time a function declares a new variable, it is "pushed" onto the stack. Then every time a function exits, all of the variables pushed onto the stack by that function are free (that is to say they are deleted). Memory is managed for the user -- not need to declare memory or free it when you no longer need it.
- Stack grows and shrinks as functions push and pop local variables
- No need to manage the memory yourself, variables are allocated and free automatically
- Stack has size limits
- Stack variables only exist while the function that created them is running
- The heap is a region of your computer's memory that is not managed automatically for you and is not as tightly managed by the CPU. More free floating region of memory and is larger. Must manually allocate and release memory -- if not a memory leak will occur, where the memory on the heap will still be set aside
- Heap has very fast access
- Don't have to explicitly de-allocate variables
- Space is efficiently managed by the CPU, memory will not become fragmented
- Local variables only, limit ons tack size, variables cannot be resized

Recursion:

- Recursion is the process in which a function calls itself directly or indirectly and the corresponding function is called a recursive function
- Using recursive algorithm, certain problems can be solved quite easily -- Tower of Hanoi
- Has a base condition and a recursive call
- If base case is not defined, stack overflow can occur

Dynamic Programming:

- Method for solving a complex problem by breaking it down into a collection of simpler subproblems, solving each of those subproblems just once, and storing their solutions using a memory-based data structure (array, map, etc.)

Big O Time & Space:

- The language that we use for talking about how long an algorithm takes to run.
- How we compare the efficiency of different approaches to a problem
- Contingent on the data structure(s) and the design of the algorithm

https://www.tutorialspoint.com/software_engineering/software_engineering_interview_questions.htm

Algorithm Videos:

- https://www.youtube.com/channel/UC6Aa5t0vHN8uj_BCbgrRZcQ
- https://www.toptal.com/c-plus-plus/interview-questions