Assignment: Formulate a problem which can be solve using any of the following algorithm and it need use any of the following data structure.

Data Structure:

1. Stacks - expression evaluation, UNDO/REDO operations in word processors, sub-routine calls, static memory allocation
2. Queues - queues are used mostly in operating systems in order to execute process scheduling
3. Arrays - are generally used for implementation of Stacks, Queues, Priority Queues, Adjacency Matrix, Heaps, Trees, etc when the allocation size is predetermined.
4. Linked Lists - these are advantageous over arrays as they can be appended ON-THE-GO and do not need any predetermined size.
5. Heaps / Priority Queues - dynamic memory allocation and priority scheduling.
6. Trees - used in compiler design for parsing of the syntax
7. Graphs - have immense usage in real life like implementation of Routing Protocols, network communication design, circuit design, sequencing the build files of dependencies

Algorithms: -

1. Dijkstra's Algorithm - used in implementation of Routing protocols for fast data packet transfer
2. Binary Search - used to find the square-root of any real number, used in fast search procedures where solution is expected to have only TWO possibilities, used in implementation of git bisect. Some data structures derived from Binary search are Binary Search Trees, SET, MAP, etc
3. Sorting Algorithms - Merge Sort, Heap Sort, Quick Sort, Selection Sort, insertion sort have their own usages in particular cases.
4. DFS/BFS - these search methods are implemented in search methods of social networks like finding of people by linking with mutual friends, etc.
5. Topological Sort - topological sort works only on DAG(directed acyclic graphs). This is mainly implemented in BUILD-SYSTEMS of IDE(s) in order to generate a proper dependency graph of the dependent modules.
6. Prim's and Kruskal's Algorithms for MST - One example would be a telecommunications company laying cable to a new neighborhood. If it is constrained to bury the cable only along certain paths (e.g. along roads), then there would be a graph representing which points are connected by those paths. Some of those paths might be more expensive, because they are longer, or require the cable to be buried deeper; these paths would be represented by edges with larger weights. A *spanning tree* for that graph would be a subset of those paths that has no cycles but still connects to every house; there might be several spanning trees possible. A *minimum spanning tree* would be one with the lowest total cost, thus would represent the least expensive path for laying the cable.