1. Design Techniques

It provides a logical and systematic means of proceeding with the design process as well as a set of guidelines for decision-making. The design methodology provides a sequence of activities, and often uses a set of notations or diagrams [1]. The design process involves developing a conceptual view of the system, establishing system structure, identifying data streams and data stores, decomposing high level functions into sub functions, establishing relationships and interconnections among components, developing concrete data representations, and specifying algorithmic details [2].

2. Divide and Conquer

It breaks a problem into sub-problems that are similar to the original problem, recursively solves the sub-problems, and finally combines the solutions to the sub-problems to solve the original problem [3]. Because divide-and-conquer solves sub-problems recursively, each sub-problem must be smaller than the original problem, and there must be a base case for sub-problems [4].

3. Greedy Method

A greedy algorithm is a simple, intuitive algorithm that is used in optimization problems. The algorithm makes the optimal choice at each step as it attempts to find the overall optimal way to solve the entire problem [5]. Greedy algorithms are quite successful in some problems, such as Huffman encoding which is used to compress data, or Dijkstra's algorithm, which is used to find the shortest path through a graph [6].

4. Dynamic Programming

Dynamic Programming (DP) is an algorithmic technique for solving an optimization problem by breaking it down into simpler sub-problems and utilizing the fact that the optimal solution to the overall problem depends upon the optimal solution to its sub-problems [7]. Dynamic Programming is mainly an optimization over plain recursion. Wherever we see a recursive solution that has repeated calls for same inputs, we can optimize it using Dynamic Programming. The idea is to simply store the results of sub-problems, so that we do not have to re-compute them when needed later [8].

5. Backtracking

Backtracking is an algorithmic-technique for solving problems recursively by trying to build a solution incrementally, one piece at a time, removing those solutions that fail to satisfy the constraints of the problem at any point of time [9]. The Backtracking is an algorithmic-method to solve a problem with an additional way. It uses a recursive approach to explain the problems. We can say that the backtracking is needed to find all possible combination to solve an optimization problem. Backtracking is a systematic way of trying out different sequences of decisions until we find one that works [10].

6. Branch & Bound

Branch and bound is an algorithm design paradigm which is generally used for solving combinatorial optimization problems. These problems are typically exponential in terms of time complexity and may require exploring all possible permutations in worst case [11]. The Branch and Bound Algorithm technique solves these problems relatively quickly. It is not a solution technique specifically limited to integer programming problems. It is a solution approach that can be applied to a number of different types of problems [12].

7. Linear Programming

Linear programming, mathematical modeling technique in which a linear function is maximized or minimized when subjected to various constraints [13]. This technique has been useful for guiding quantitative decisions in business planning, in industrial engineering, and to a lesser extent in the social and physical sciences [14].

8. Mesh Network

A mesh network is a group of devices that act as a single Wi-Fi network. So, there are multiple sources of Wi-Fi around your house, instead of just a single router. These additional Wi-Fi sources are called points [15]. Mesh Wi-Fi uses multiple nodes to create a single, big and seamless Wi-Fi network that covers your whole home. Your devices will connect to the closest node automatically and without any disruptions as you move around your house [16].

9. Matrix Multiplication

In mathematics, particularly in linear algebra, matrix multiplication is a binary operation that produces a matrix from two matrices. For matrix multiplication, the number of columns in the first matrix must be equal to the number of rows in the second matrix [17]. The resulting matrix, known as the matrix product, has the number of rows of the first and the number of columns of the second matrix. The product of matrices A and B is denoted as AB [18].

10. Hypercube Network

In computer networking, hypercube networks are a type of network topology used to connect multiple processors with memory modules and accurately route data. Hypercube networks consist of 2m nodes, which form the vertices of squares to create an inter-network connection [19]. A hypercube is basically a multidimensional mesh network with two nodes in each dimension. Due to similarity, such topologies are usually grouped into a k-ary d-dimensional mesh topology family, where d represents the number of dimensions and k represents the number of nodes in each dimension [20].

11. Block Matrix

In mathematics, a block matrix or a partitioned matrix is a matrix that is interpreted as having been broken into sections called blocks or submatrices [21]. Intuitively, a matrix interpreted as a block matrix can be visualized as the original matrix with a collection of horizontal and vertical lines, which break it up, or partition it, into a collection of smaller matrices. Any matrix may be interpreted as a block matrix in one or more ways, with each interpretation defined by how its rows and columns are partitioned [22].

12. Enumeration Sort

Enumeration sort is a method of arranging all the elements in a list by finding the final position of each element in a sorted list. It is done by comparing each element with all other elements and finding the number of elements having smaller value [23]. Also, t is a method of finding the exact position of each element in a sorted list by comparing and finding the frequency of elements having smaller value [24].

13. Odd-Even Transposition Sort

Odd-Even Transposition Sort is a parallel sorting algorithm. It is based on the Bubble Sort technique, which compares every 2 consecutive numbers in the array and swap them if first is greater than the second to get an ascending order array. It consists of 2 phases the odd phase and even phase [25]. It is a relatively simple sorting algorithm, developed originally for use on parallel processors with local interconnections [26].

14. Parallel Merge Sort

A parallel merge algorithm performs a merge operation on two sorted sequences of length, each distributed over tasks, to produce a single sorted sequence of length distributed over tasks [27]. In this algorithm, the data is split in half in the merge sort function, which is then called again on each half. This is done recursively until the size of each 'half' is one, at which point the left and right halves are merged together in a sorted list using the merge function. As each recursive call to merge completes, more of the halves are merged in sorted order and stored in a new array called result [28].

15. Hyper Quick Sort

Hyper quick sort is an implementation of quick sort on hypercube. We Divide the unsorted list among each node [29]. Then Sort each 'I' node locally. From node 0, broadcast the median value. And split each list locally, then exchange the halves across the highest dimension [30].

16. Depth-First Search

Depth-first search (DFS) is an algorithm for traversing or searching tree or graph data structures. The 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 [31]. It traverses a graph in a depth ward motion and uses a stack to remember to get the next vertex to start a search, when a dead end occurs in any iteration [32].

17. Breadth-First Search

Breadth-first search is an algorithm for searching a tree data structure for a node that satisfies a given property. It starts at the tree root and explores all nodes at the present depth prior to moving on to the nodes at the next depth level [33]. The only catch here is, unlike trees, graphs may contain cycles, so we may come to the same node again. To avoid processing a node more than once, we use a Boolean visited array. For simplicity, it is assumed that all vertices are reachable from the starting vertex [34].

18. Best-First Search

Best-first search is a search algorithm which explores a graph by expanding the most promising node chosen according to a specified rule [35]. The idea of Best First Search is to use an evaluation function to decide which adjacent is most promising and then explore. Best First Search falls under the category of Heuristic Search or Informed Search [36].

19. Binary search

Binary search is an efficient algorithm for finding an item from a sorted list of items. It works by repeatedly dividing in half the portion of the list that could contain the item, until you've narrowed down the possible locations to just one [37]. Search a sorted array by repeatedly dividing the search interval in half. Begin with an interval covering the whole array. If the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half. Otherwise, narrow it to the upper half. Repeatedly check until the value is found or the interval is empty [38].

20. Graph Coloring

A graph coloring is an assignment of labels, called colors, to the vertices of a graph such that no two adjacent vertices share the same color. The chromatic number of a graph GGG is the minimal number of colors for which such an assignment is possible. Other types of coloring's on graphs also exist, most notably edge coloring's that may be subject to various constraints [39]. In graph theory, graph coloring is a special case of graph labeling; it is an assignment of labels traditionally called "colors" to elements of a graph subject to certain constraints. By planar duality it became coloring the vertices, and in this form, it generalizes to all graphs [40].

21. Minimal Spanning Tree

A minimum or minimal spanning tree or minimum weight spanning tree is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight [41]. That is, it is a spanning tree whose sum of edge weights is as small as possible [42].

22. Prim’s Algorithm

Prim's algorithm is a minimum spanning tree algorithm that takes a graph as input and finds the subset of the edges of that graph which form a tree that includes every vertex and has the minimum sum of weights among all the trees that can be formed from the graph [43]. Prim's algorithm, in contrast with Kruskal's algorithm, treats the nodes as a single tree and keeps on adding new nodes to the spanning tree from the given graph. Prim's algorithm to find minimum cost spanning tree (as Kruskal's algorithm) uses the greedy approach. Prim's algorithm shares a similarity with the shortest path first algorithms [44].

23. Kruskal's Algorithm

Kruskal's algorithm to find the minimum cost spanning tree uses the greedy approach. This algorithm treats the graph as a forest and every node it has as an individual tree [45]. A tree connects to another only and only if, it has the least cost among all available options and does not violate MST properties [46].

24. Shortest Path Algorithm

The shortest-path algorithm. The shortest-path algorithm calculates the shortest path from a start node to each node of a connected graph. It is the basis for all the apps that show you a shortest route from one place to another [47]. The algorithm exists in many variants. Dijkstra's original algorithm found the shortest path between two given nodes, but a more common variant fixes a single node as the "source" node and finds shortest paths from the source to all other nodes in the graph, producing a shortest-path tree [48].

25.Moore's algorithm

The Boyer-Moore algorithm is considering the most efficient string-matching algorithm in usual applications, for example, in text editors and commands substitutions. The reason is that its woks the fastest when the alphabet is moderately sized and the pattern is relatively long [49]. Pattern searching is an important problem in computer science. When we do search for a string in a notepad/word file, browser, or database, pattern searching algorithms are used to show the search results [50].

References

[1] Design Technique - https://ecomputernotes.com/software-engineering/explain-various-design-techniques
[2] Design Technique - https://userpages.umbc.edu/~khoo/survey1.html
[3] Divide and Conquer - https://www.tutorialspoint.com/data_structures_algorithms/divide_and_conquer.htm
[4] Divide and Conquer - https://www.khanacademy.org/computing/computer-science/algorithms/merge-sort/a/divide-and-conquer-algorithms
[5] Greedy Algorithm - https://www.geeksforgeeks.org/greedy-algorithms/
[6] Greedy Algorithm - https://brilliant.org/wiki/greedy-algorithm/
[7] Dynamic Programming - https://www.geeksforgeeks.org/dynamic-programming/
[8[Dynamic Programming - https://www.educative.io/courses/grokking-dynamic-programming-

patterns-for-coding-interviews/m2G1pAq0OO0
[9] Backtracking - https://www.javatpoint.com/backtracking-introduction
[10] Backtracking - https://www.geeksforgeeks.org/backtracking-introduction/
[11] Branch & Bound -
http://web.tecnico.ulisboa.pt/mcasquilho/compute/_linpro/TaylorB_module_c.pdf
[12] Branch & Bound - https://www.geeksforgeeks.org/branch-and-bound-algorithm/
[13] Linear Programming - https://en.wikipedia.org/wiki/Linear_programming
[14] Linear Programming - https://www.britannica.com/science/linear-programming-mathematics
[15] Mesh Network - https://eu.dlink.com/uk/en/resource-centre/blog/wifi-extender-vs-mesh-wifi-which-is-better
[16] Mesh Network - https://support.google.com/wifi/answer/7182746?hl=en
[17] Matrix Multiplication - https://www.varsitytutors.com/hotmath/hotmath_help/topics/matrix-multiplication
[18] Matrix Multiplication - https://en.wikipedia.org/wiki/Matrix_multiplication
[19] Hypercube Network - https://www.sciencedirect.com/topics/computer-science/hypercube

[20] Hypercube Network - https://www.osapublishing.org/ao/abstract.cfm?uri=ao-33-32-7588
[21] Block Matrix - https://ximera.osu.edu/la/LinearAlgebra/MAT-M-0023/main

[22] Block Matrix - https://en.wikipedia.org/wiki/Block_matrix
[23] Enumeration Sort -
https://www.tutorialspoint.com/parallel_algorithm/parallel_algorithm_sorting.htm
[24] Enumeration Sort - https://ecs073pa.files.wordpress.com/2014/11/lecture-iit-2.pdf
[25] Odd-Even Transposition Sort - https://en.wikipedia.org/wiki/Odd%E2%80%93even_sort
[26] Odd-Even Transposition Sort - https://www.geeksforgeeks.org/odd-even-transposition-sort-brick-sort-using-pthreads/
[27] Parallel Merge Sort -
http://selkie.macalester.edu/csinparallel/modules/ParallelSorting/build/html/MergeSort/MergeSort.html
[28] Parallel Merge Sort - https://www.mcs.anl.gov/~itf/dbpp/text/node127.html
[29] Hyper Quick Sort - https://cse.buffalo.edu/faculty/miller/Courses/CSE633/Srivastava-Spring-2014-CSE633.pdf
[30] Hyper Quick Sort -
https://www.lkouniv.ac.in/site/writereaddata/siteContent/202005011807550420Rohit-Srivastava-hyper_quick_sort.pdf
[31] Depth-First Search -
https://www.tutorialspoint.com/data_structures_algorithms/depth_first_traversal.htm
[32] Depth-First Search - https://en.wikipedia.org/wiki/Depth-first_search
[33] Breadth-First Search - https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/
[34] Breadth-First Search - https://en.wikipedia.org/wiki/Breadth-first_search
[35] Best-First Search - https://www.geeksforgeeks.org/best-first-search-informed-search/

[36] Best-First Search - https://www.mygreatlearning.com/blog/best-first-search-bfs/
[37] Binary Search - https://www.geeksforgeeks.org/binary-search/

[38] Binary Search - https://www.khanacademy.org/computing/computer-science/algorithms/binary-search/a/binary-search
[39] Graph Coloring - https://en.wikipedia.org/wiki/Graph_coloring

[40] Graph Coloring - https://brilliant.org/wiki/graph-coloring-and-chromatic-numbers/
[41] Minimal Spanning Tree - https://www.hackerearth.com/practice/algorithms/graphs/minimum-spanning-tree/tutorial/

[42] Minimal Spanning Tree - https://www.statisticshowto.com/minimum-spanning-tree/
[43] Prim Algorithm - https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/

[44] Prim Algorithm - https://www.tutorialspoint.com/data_structures_algorithms/prims_spanning_tree_algorithm.htm
[45] Kruskal Algorithm - https://www.programiz.com/dsa/kruskal-algorithm
[46] Kruskal Algorithm - https://www.tutorialspoint.com/data_structures_algorithms/kruskals_spanning_tree_algorithm.htm

[47] Shortest Path Algorithm - https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/

[48] Shortest Path Algorithm- https://www.cs.cornell.edu/courses/JavaAndDS/shortestPath/shortestPath.html

[49] Moore's algorithm - https://www.geeksforgeeks.org/boyer-moore-algorithm-for-pattern-searching/

[50] Moore's algorithm - http://personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/StringMatch/boyerMoore.htm

Honor Pledge

"I affirm that I have not given or received any unauthorized help on this activity, and this work is my own."

- Eustaquio, Aron Clark R.