Searching is one of the fundamental operations in computer science. It is used in all applications where we need to find if an element is in the given list or not. In this chapter, we will discuss the following search algorithms –

- Divide and Conquer
- Depth-First Search
- Breadth-First Search
- Best-First Search

**Divide and Conquer**

In divide and conquer approach, the problem is divided into several small sub-problems. Then the sub-problems are solved recursively and combined to get the solution of the original problem.

The divide and conquer approach involves the following steps at each level –

- **Divide** – The original problem is divided into sub-problems.
- **Conquer** – The sub-problems are solved recursively.
- **Combine** – The solutions of the sub-problems are combined to get the solution of the original problem.

Binary search is an example of divide and conquer algorithm.

**Pseudocode**

```python
def BinarySearch(a, b, low, high):
    if low < high:
        return NOT FOUND
    else:
        mid = (low + high) / 2
        if b = key(mid) then
            return key(mid)
        else if b < key(mid) then
            return BinarySearch(a, b, low, mid-1)
        else
            return BinarySearch(a, b, mid+1, high)
```

**Depth-First Search**

Depth-First Search or DFS is an algorithm for searching a tree or an undirected graph data structure. Here, the concept is to start from the starting node known as the root and traverse as far as possible in the same branch. If we get a node with no successor node, we return and continue with the vertex, which is yet to be visited.

**Steps of Depth-First Search**

- Consider a node root that is not visited previously and mark it visited.
- Visit the first adjacent successor node and mark it visited.
- If all the successors nodes of the considered node are already visited or it doesn’t have any more successor node, return to its parent node.

**Pseudocode**
Let \(v\) be the vertex where the search starts in Graph \(G\).

Breadth-First Search

Breadth-First Search or BFS is an algorithm for searching a tree or an undirected graph data structure. Here, we start with a node and then visit all the adjacent nodes in the same level and then move to the adjacent successor node in the next level. This is also known as level-by-level search.

Steps of Breadth-First Search

- Start with the root node, mark it visited.
- As the root node has no node in the same level, go to the next level.
- Visit all adjacent nodes and mark them visited.
- Go to the next level and visit all the unvisited adjacent nodes.
- Continue this process until all the nodes are visited.

Pseudocode

Let \(v\) be the vertex where the search starts in Graph \(G\).

Best-First Search

Best-First Search is an algorithm that traverses a graph to reach a target in the shortest possible path. Unlike BFS and DFS, Best-First Search follows an evaluation function to determine which
node is the most appropriate to traverse next.

**Steps of Best-First Search**

- Start with the root node, mark it visited.
- Find the next appropriate node and mark it visited.
- Go to the next level and find the appropriate node and mark it visited.
- Continue this process until the target is reached.

**Pseudocode**

```plaintext
BFS( m )
    Insert( m.StartNode )
    Until PriorityQueue is empty
        c ← PriorityQueue.DeleteMin
        If c is the goal
            Exit
        Else
            Foreach neighbor n of c
                If n "Unvisited"
                    Mark n "Visited"
                    Insert( n )
            Mark c "Examined"
End procedure
```

Loading [Mathjax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js