Title -

Design and implement parallel breadth first search and depth first search based on existing algorithms using openMP. Use a tree or and undirected graph for BFS & DFS.

Objective -

- To understand the parallel search algorithm.
- To implement the DFS & BFS search algorithm.

Problem Statement -

Design and implement parallel BFS & DFS based on the existing algorithms using OpenMP.

Software & Hardware Requirements -

- Multicore processor, 4GB RAM, 500 GB HDD.
- 64 bit open source windows or linux
- Programming languages : Python / C++.

Theory -

Parallel Algorithm -

The breadth-first-search is a way to explore the vertices of a graph layer by layer. It is a basic algorithm in graph theory which can be used as a part of other graph algorithm. 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. Following are some search algorithms:

- Divide & conquer
- Depth - first search.
- Breadth - first search.
- Best - first search.

Breadth - First Search. -

Breadth-first Search (BFS) is an algorithm for searching a tree or an undirected graph

data structure. Here, we start with a node & then visit all the adjacent nodes in the same level & 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** -

1. Start with the root node, mark it visited.

2. As the root node has no node in the same level, go to the next level.

3. Visit all adjacent nodes & mark them visited.

4. Go to the next level & visit all the unvisited adjacent nodes.

5. Continue this process until all the nodes are visited.

Pseudocode -

```
BFS (G, v)
    Queue Q := { };

    for each vertex u, set visited [u] := false;
    insert Q, v;
    while (Q is not empty) do
        u := delete Q;

        if (not visited [u]) then
            visited [u] := true;
            for each unvisited neighbor w of u
                insert Q, w;

    end if

    end while.

END BFS ()
```

Depth - First - Search -

Depth - First Search (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 & traverse as far as possible in the same branch. If we get a node with no successor node, we return & continue with the vertex, which is yet to be visited.

- ## Steps of Depth - First Search -

1. Consider a node (root) that is not visited previously & mark it visited.

2. Visit the first adjacent successor node & mark it visited.

3. 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 –

```
DFS (G, v)

    Stack S := { };

    for each vertex u, set visited [u] := false;
    push S, v;
    while (S is not empty) do.
        u := pop S;

        if (not visited [u]) then.
            visited [u] := true;
            for each unvisited neighbor w of u
                push S, w;

        endif

    end while.

END DFS ().
```

Conclusion -

Hence, we have studied the implemention of parallel algorithm for BFS & DFS search algorithms.