Assignment No. 01

Title: Design and implement Parallel Breadth-First search and Depth-First Search based on existing algorithm using openMP. Use a tree or an unidirected graph for BFS & DFS

Objective: students should be able to write a program to implement parallel BFS & DFS based on existing algorithms using openMP.

Theory:

Breadth First search (BFS):

BFS stands for Breadth First Search. It is a graph traversal algorithm used to explore all the nodes of a graph or tree systematically, starting from the root node or a specified starting point and visiting all the neighboring node of current depth level before moving on to the next depth level.

The algorithm uses a queue data structure to keep track of the nodes that need to be visited, and marks each visited node to avoid processing it again.

The basic idea of the BFS algorithm is visit all the nodes at a given level of moving on to the next level, which ends that all the nodes are visited in BFS.

Example of BFS

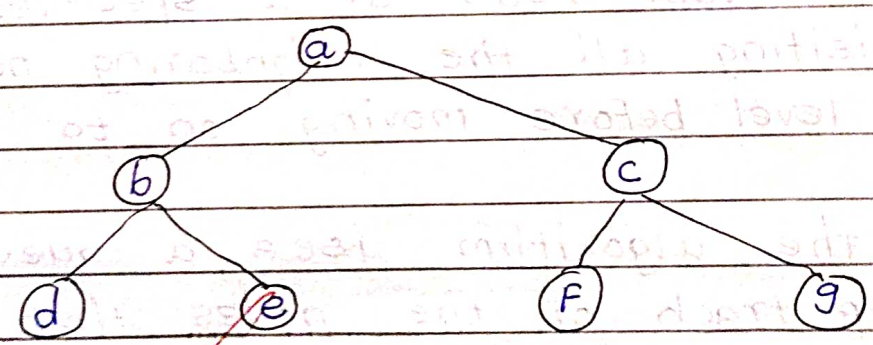
Now let's take a look at the step in traversing a graph by using BFS.

step 1 - Take a empty queue.

step 2 - select a starting node and inse queue.

step 3 - provided that the queue is not the node from the queue and i child nodes into the queue

step 4 print the extracted node



queue

| | | | | | | | a |

Print a

| | | | | | | c | b |

print a insert its child nodes int

Print b

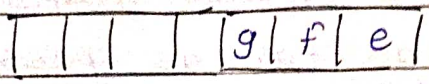
| | | | | | e | d | c |

Print b and insert its child node i

Print c

print 'c' & insert node into the queue

print d

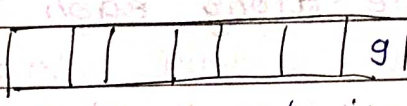


print 'd' & insert its child node into the queue

print e

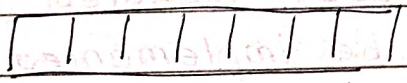
print 'e' & insert its child node into the que

print f



print 'f' & insert its child node into queue

print g



print 'g' & insert its child nodes into queue

OpenMP :

openmp (open multi-processing) is an api that support shared memory parallel programming in c, c++ and Fortron.

It is used to write parallel program that can run on multiple processor system and parallel computing cluster.

openmp provides a set of directives and that can be inserted into the source of a program to parallelize its execution.

These directives are simple and easy to use and they can be applied to loops, section, function and other program construct.

Depth First Search (DFS):

DFS stands for Depth First Search
It is a popular Graph traversal alon
explorer as far as possible along bra
backtracking.

This algorithm can be used to find
path between two vertices or traverse
a systematic way.

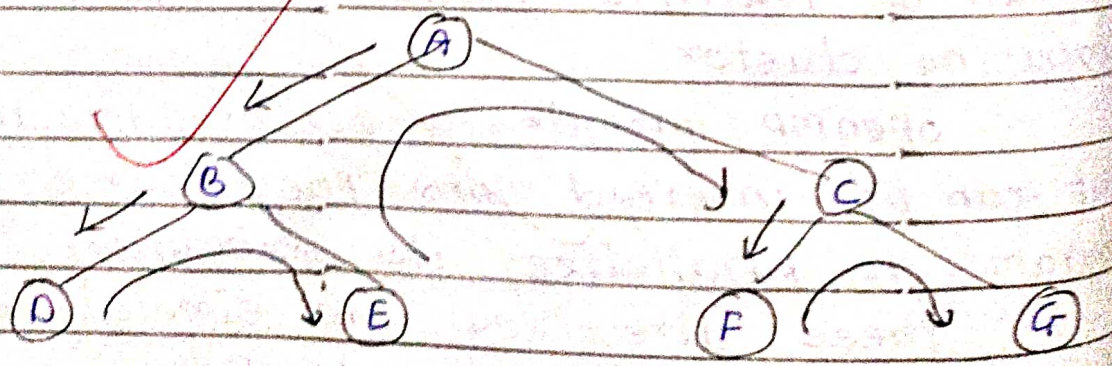
The algorithm starts at root node an
far as possible along each branch befo
tracking.

The backtracking is done to explore
that has not been explored yet.

DFS can be implemented using eith
or an iterative approach.

The recursive approach is simpler to
can lead to a stack overflow for
large graphs.

The iterative approach uses a sta
keep track of node to be explored and
for large graphs.



The purpose of the algorithm is to mark each vertex as visited while avoiding cycles.

To implement DFS Traversal, you need to take the following stages.

1 - create a stack with the total no. of vertices in the graph as the size

2 - choose any vertex or the traversal's begining point, push a visit to that vertex and add it to the stack.

3 - push any non visited adjacent vertices of a vertex at the top of the stack to the top of the stack.

4 - Repeat step 3 & 4 until there are no more vertices to visit from the vertex at the top of the stack.

5 - If there are no new vertices to visit back and pop one from the stack backtrack

6 - continue using step 3, 4 and 5 until stack is empty.

7 - when the stack is entirely u
the final spanning tree by deleti
unused edges

Conclusion -

In this way, we can achie
while implementing BFS and DFS