Page No.:
Date:

# Tutorial - 5

1) difference between D.F.S and B.F.S

| (a) DFS | B.FS |
|---|---|
| → D.F.S stands for depth first search | B.F.S stands for Breadth first search |
| → Uses stack data structure | Uses queue data structure |
| → DFS is more suitable when there are solutions away from source | B.F.S is more suitable when searching vertices are closer to source. |
| → Time complexity is O(V+E) with adjacency matrix is used and $O(V^2)$ without adjacency matrix. | T.C of B.F.S is (O(V+E) with adjacency matrix and $O(V^2)$ without adjacency matrix. |

→ D.F.S applications.

(a) if D.F.S is applied to an unweighted graph then it creates a minimum spanning tree for all pair shortest path.

(b) D.F.S can be used to detect cycles in graph.

(c) D.F.S can be used to find path between 2 vertices.

→ B.F.S applications.

(a) It can be used to find shortest path between 2 nodes in an unweighted graph.

(b) G.P.S technology uses B.F.S to find neighbouring places.

Page No.:
Date:
YOUVA

2- Stack is used in D.F.S because in D.F.S we first traverse the whole branch of the tree and later on visit the adjacent branch, since this is similar to LIFO, therefore stack is used to implement D.F.S.

Queue is used in B.F.S, it is because queue is used as a FIFO data structure, instead of stack because the sequence for B.F.S is to visit the immediate children first, and after all immediate children are visited, then return to those children and check their children's and so on.

3- Sparse Graph: A graph where number of edges is much less than possible number of edges.

Dense Graph: A graph where no. of edges is much close to maximum number of edges

- If a graph is dense it should be represented by adjacency matrix
- If a graph is sparse it should be represented by adjacency matrix

4- Cycle detection using B.F.S

→ do a B.F.S traversal on given graph, for each visited vertex v, if there is an adjacent 'u' such that 'u' is already visited and 'u' is not parent of 'v', then there is a cycle in graph.

Cycle detection using D.F.S

→ run D.F.S from a node and mark it as visited now for any other vertex vertex if its neighbour

is already visited and that parent neighbour is not the parent of that current node then there exists a cycle of the in the graph.

(5) Disjoint set Data Structure :-
- The disjoint set can be defined as the subsets where there is no common element between two sets.

Operations are :-
(a) union
(b) new set
(c) find

(6) BFS → A → B → C → D → E → F
DFS → A → B → E → C → D → F

(7) Connected component → 4
vertices → 10

(8) Topological sort → 0 - 1 - 2 - 3 - 4 - 5
D.F.S → 5 → 2 → 3 → 1 → 0
4 can't be reached.

(9) Yes, heap ~~sort~~ data structure can be used to create priority queue
→ dijkstra's algorithm to find shortest path
→ prim's algorithm
→ Hauffman algorithm

(10) min heap — root element is the shortest
max heap — root element is the largest.