

Tutorial - 5

Ans-1 BFS \rightarrow It stands for Breadth first search. It uses queue to find the shortest path. It is better when target is closer to source. It considers all neighbours so it is not suitable for decision tree used in Puzgel game. It is slower than DFS. $TC \rightarrow O(V+E)$.

DFS \rightarrow It stands for Depth first search. It uses stack to find the shortest path. It is better when target is far from source. It is more suitable as with one decision. We need to traverse far further to argument the decision. It is faster than BFS. $TC \rightarrow O(V+E)$.

Ans-2 Stack is used to implement DFS, because it is 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.

Queue is used to implement BFS, it is because queue uses FIFO instead because BFS is to test the immediate children first and after all immediate children are tested, to their. Return to those children & check their children & so forth.

Ans-3 Sparse Graph \rightarrow Graph where number of edge is much less than the possible number of edges.

Dense Graph \rightarrow Graph where number of edges is much more than close to maximal number of edges.

- if Graph is dense it should be represented by adjacency matrix.
- if Graph is sparse it should be represented by adjacency list.

Ans-4 BFS \rightarrow In undirected graph, do a BFS traversal on given graph, for each visited vertex v , if there is any adjacent ' u ' such that ' v ' is already visited & ' u ' is not Parent of ' v ' then there is Cycle in a graph.

DFS

Run DFS from a node and mark that node as visited now for any other vertices if its neighbour is already visited & that neighbour is not its Parent then there exists a cycle in graph.

Ans-5 Disjoint Set data structure

The disjoint set can be defined as the sub set where there is no common element between two set operation are
1 Union, 2 make new set, 3 find

Ans-6 BFS

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$

$G \rightarrow H \rightarrow F$

DFS

$A \rightarrow D \rightarrow C \rightarrow B$

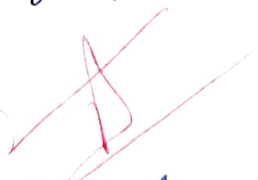
$G \rightarrow F \rightarrow H$

Ans-7 Connected component $\rightarrow 4$, Vertices $\rightarrow 10$

Ans-8 Topological sort $\rightarrow 0-1-2-3-4-5$, DFS: $5 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 0$

Ans-9 Yes heap DS can be used to make priority queue.

- 1 Dijkstra to find shortest path
- 2 Prim's algo
- 3 Huffman algo



Ans-10 Min heap \rightarrow Root element is smaller
Max heap \rightarrow Root element is largest.