**Q. Explain breadth-first search with an algorithm.**

Breadth first search is a graph traversal algorithm that starts traversing the graph from root node and explores all the neighboring nodes.

Then, it selects the nearest node and explore all the unexplored nodes. The algorithm follows the same process for each of the nearest node until it finds the goal.

BFS algorithm uses Queue (First in First out) data structure.

This strategy is often not feasible when the search space is large.

The algorithm of breadth first search is given below. There are 2 rules to follow.

1. Visit the starting vertex and marked it as visited. Display it.

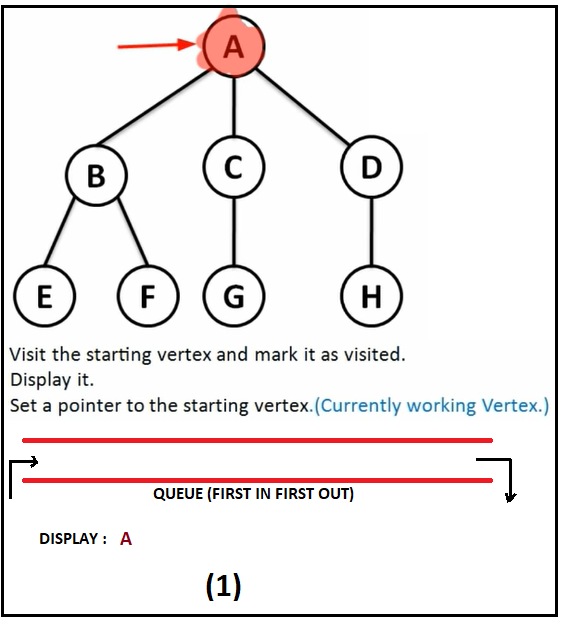
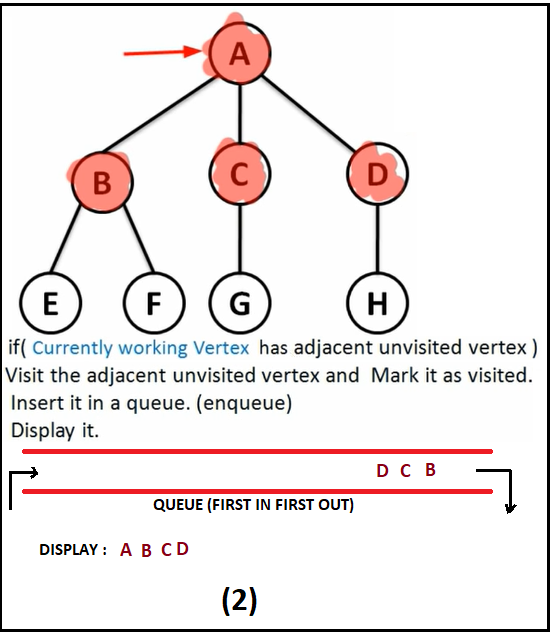
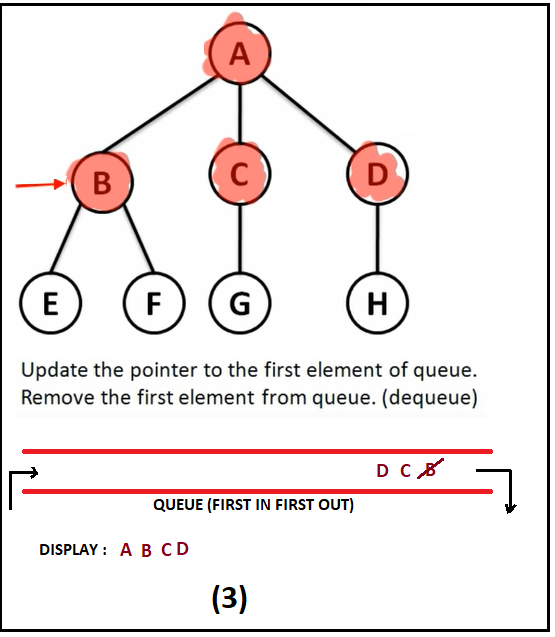
Set the pointer to the starting vertex.

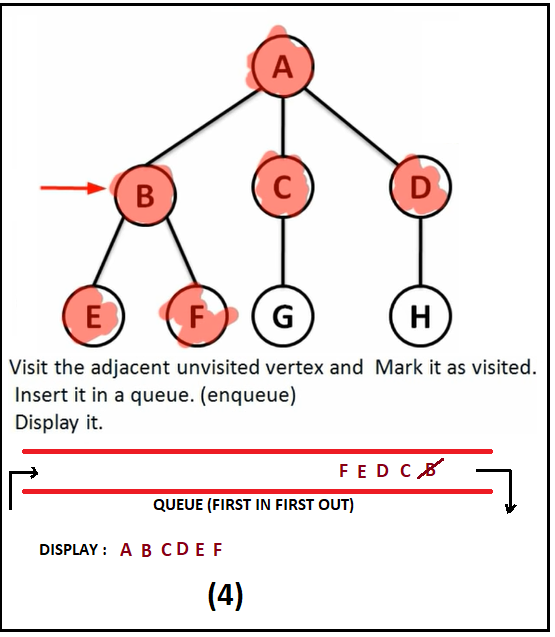
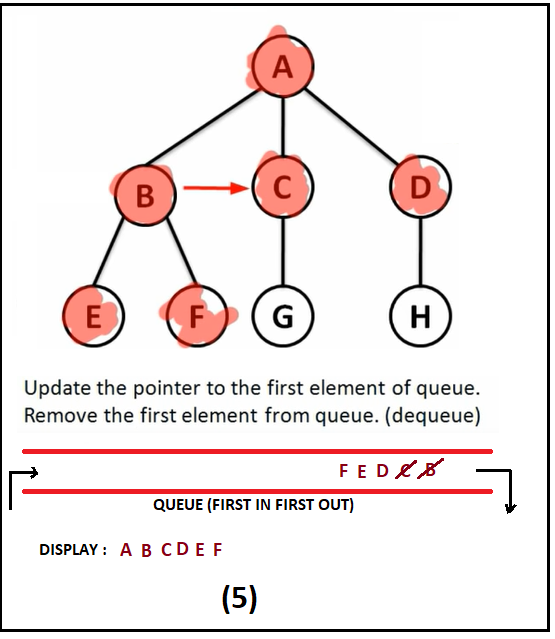
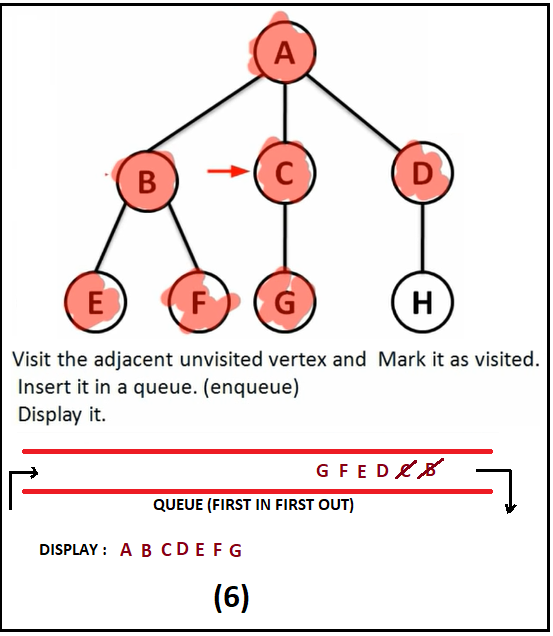
1. If the current working vertex has adjacent unvisited vertex, then visit the adjacent unvisited vertex and marked as visited. Insert in queue(enqueue). Display it.

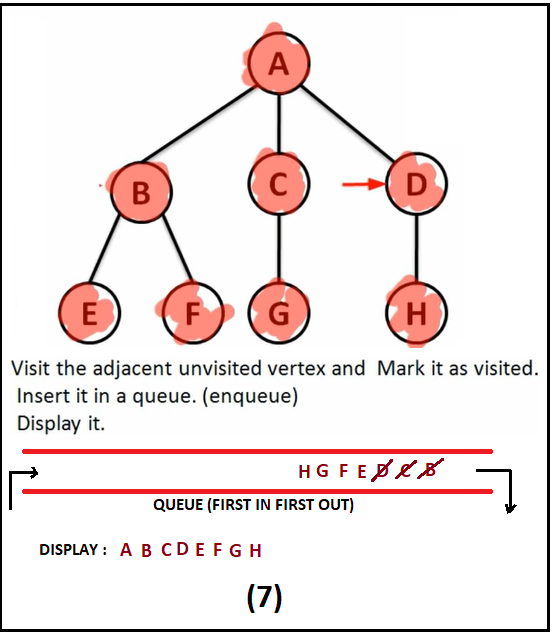
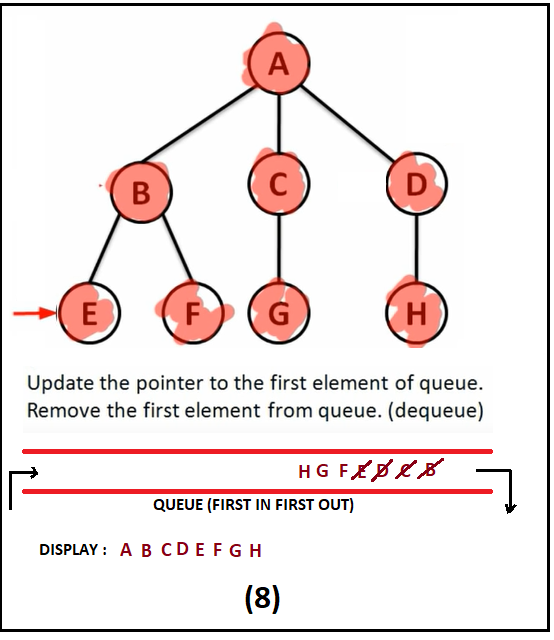
If the current working vertex has no adjacent unvisited vertex, then update the pointer to the first element of queue. Remove the first element from queue (dequeue).

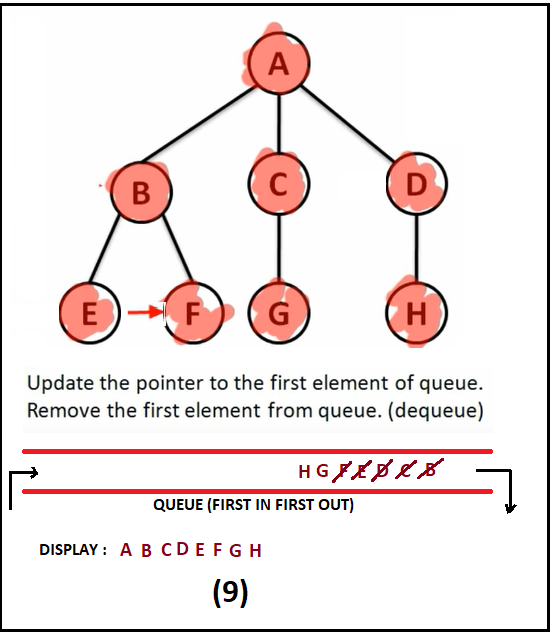
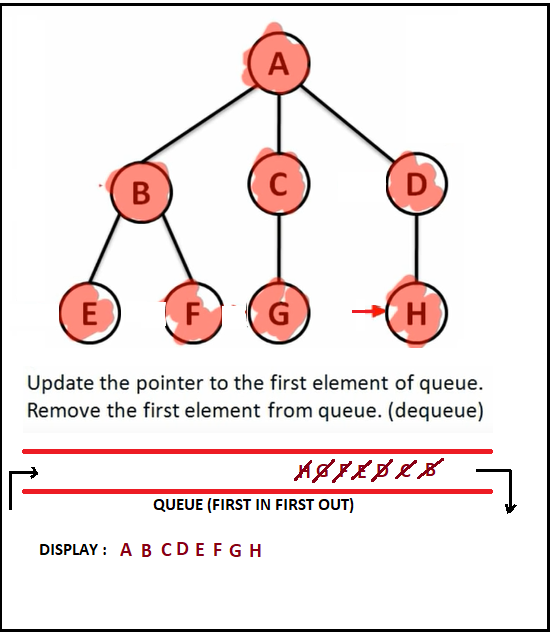
Repeat rule 2 until queue is empty.

The algorithm starts with examining the node A as a source node. In the next step, the neighbours of the nearest node of A are explored and process continues in the further steps. The algorithm explores all neighbors of all the nodes and ensures that each node is visited exactly once and no node is visited twice.

In the last step we will find the queue is empty and display the visited element.

**Advantages:**

1. Guaranteed to find an optimal solution (in terms of shortest number of steps to reach the goal).
2. Can always find a goal node if one exists (complete).

**Disadvantages:**

1. High storage requirement: exponential with tree depth.