3.2 BFS Algorithm

Steps of Breadth First Search (BFS) Algorithm

Input: A connected, undirected graph G.
Output: A spanning tree T of G.

Step 1: Initial Setup: Discard all loops and parallel edges from the undirected connected graph G.

Step 2: Choose Root Vertex: Arbitrarily choose a vertex from G and designate it as the root.

Step 3: Expand Level-1 Vertices: Add all edges incident to the root, ensuring no cycles are formed. The new vertices added here form level-1 in the spanning tree. Arbitrarily order these vertices.

Step 4: Expand to Next Level: For each vertex at level-1, in sequence, add each edge incident to it to the tree as long as no cycles are formed. Order the children of each vertex at level-1. The resulting vertices form level-2.

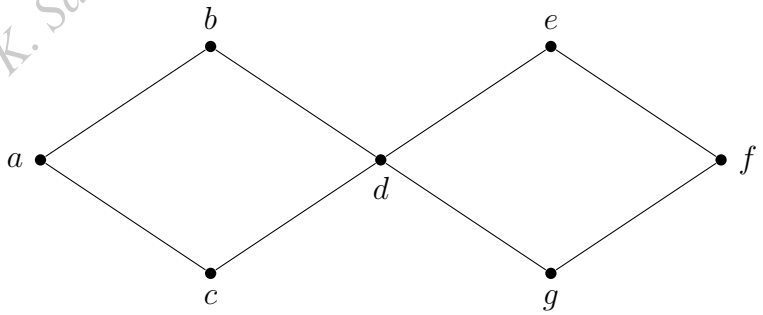
Step 5: Repeat Until Completion: Continue expanding levels in the same manner until all vertices are added to the tree.

Step 6: End Condition: The process stops after all vertices have been added, producing a spanning tree with $(n-1)$ edges for a graph G with n vertices.

Note *The time complexity of the BFS algorithm is $O(n+m)$, where n is the number of vertices and m is the number of edges in G.*

3.3 Illustrative Examples

Example 3.1 Use BFS algorithm to find a spanning tree of G:

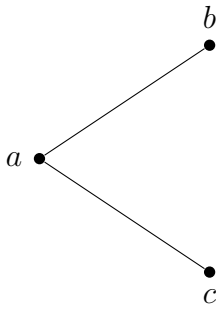


Solution *The graphical steps of BFS are shown below:*

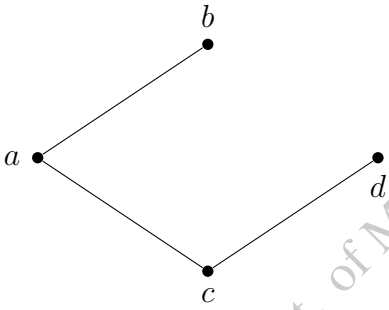
***Step** 1: Choose the vertex a to be the root.*

a •

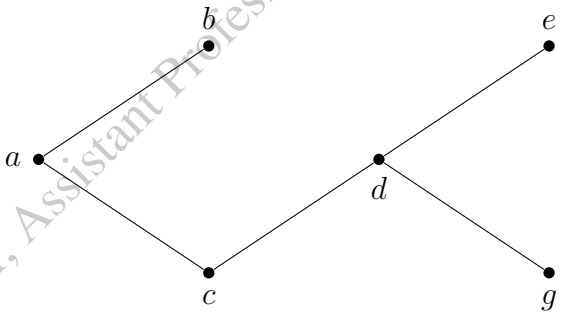
***Step** 2: Add edges incident with all vertices adjacent to a, so that edges $\{a,b\}$ and $\{a,c\}$ are added. The two vertices b and c are in level-1 in the tree.*



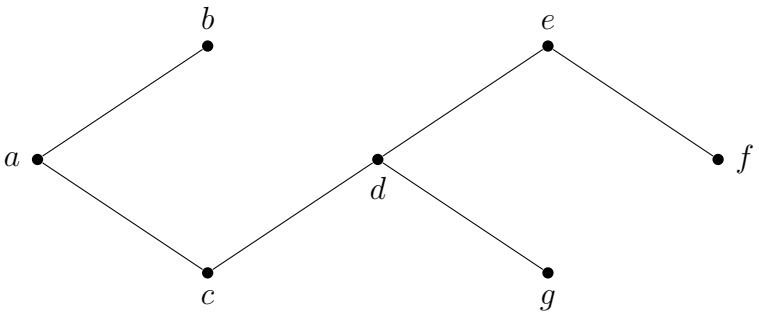
***Step** 3: Add edges from these vertices at level-1 to adjacent vertices not already in the tree (do not form a circuit). Hence the edge (c, d) is added. The vertex d is in level-2.*



***Step** 4: Add edges from d to adjacent vertices. The edges $\{d, e\}$ and $\{d, g\}$ are added. Hence e and g are in level-3.*



***Step** 5: Add edges from e to adjacent vertices, and hence $\{e, f\}$ is added.*



Required spanning Tree $T(V', E')$

Hence the final figure is the required spanning tree which contains n vertices with $(n-1)$ edges of G.

✍ **Exercise 3.1** Using BFS (Breadth First search) algorithm find the possible traversal for the above graph, if starting vertex is P.