| **Parameters** | **BFS** | **DFS** |
| --- | --- | --- |
| **Stands for** | BFS stands for Breadth First Search. | DFS stands for Depth First Search. |
| **Data Structure** | BFS(Breadth First Search) uses Queue data structure for finding the shortest path. | DFS(Depth First Search) uses Stack data structure. |
| **Definition** | BFS is a traversal approach in which we first walk through all nodes on the same level before moving on to the next level. | DFS is also a traversal approach in which the traverse begins at the root node and proceeds through the nodes as far as possible until we reach the node with no unvisited nearby nodes. |
| **Conceptual Difference** | BFS builds the tree level by level. | DFS builds the tree sub-tree by sub-tree. |
| **Approach used** | It works on the concept of FIFO (First In First Out). | It works on the concept of LIFO (Last In First Out). |
| **Suitable for** | BFS is more suitable for searching vertices closer to the given source. | DFS is more suitable when there are solutions away from source. |
| **Applications** | BFS is used in various applications such as bipartite graphs, shortest paths, etc. If weight of every edge is same, then BFS gives shortest path from source to every other vertex. | DFS is used in various applications such as acyclic graphs and finding strongly connected components etc. There are many applications where both BFS and DFS can be used like Topological Sorting, Cycle Detection, etc. |

