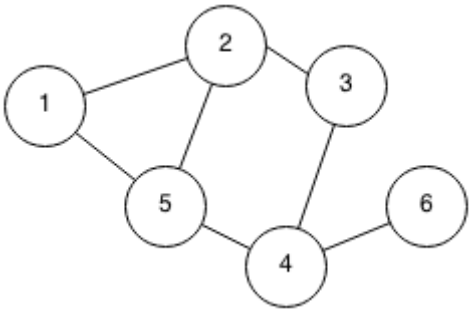
Breadth First Search vs. Depth First Search

Chadi Hani – 17105697

Arrays a can be used to create data structures other than the classes like linked lists, trees, and graphs. In this report , we will be focusing on graphs. We will cover the two most common graph search algorithms: Breadth First Search and Depth First Search.

Breadth First Search

BFS refers to the method in which you traverse a graph by visiting all children of a node before moving on to that child's children. You can think in terms of levels. If the root node is Level 1, you visit all of Level 2, then all of Level 3, all of Level 4, and so on and so forth.



Depth First Search

Whereas BFS goes level by level, DFS follows the path of one child as far as it can go, from root to finish, before going back and starting down the path of another child.

	BFS	DFS
1.	BFS stands for Breadth First Search.	DFS stands for Depth First Search.
2.	BFS(Breadth First Search) uses Queue data structure for finding the shortest path.	DFS(Depth First Search) uses Stack data structure.
3.	BFS can be used to find single source shortest path in an unweighted graph, because in BFS, we reach a vertex with minimum number of edges from a source vertex.	In DFS, we might traverse through more edges to reach a destination vertex from a source.
3.	**BFS is more suitable for searching vertices which are closer to the given source.**	**DFS is more suitable when there are solutions away from source.**
4.	**BFS considers all neighbors first and therefore not suitable for decision making trees used in games or puzzles.**	**DFS is more suitable for game or puzzle problems. We make a decision, then explore all paths through this decision.**

		And if this decision leads to win situation, we stop.
5.	The Time complexity of BFS is O(V + E) when Adjacency List is used and O(V^2) when Adjacency Matrix is used, where V stands for vertices and E stands for edges.	The Time complexity of DFS is also O(V + E) when Adjacency List is used and O(V^2) when Adjacency Matrix is used, where V stands for vertices and E stands for edges.