**Notes on** [**Paper**](https://www.researchgate.net/publication/358642884_Comparison_Analysis_of_Breadth_First_Search_and_Depth_Limited_Search_Algorithms_in_Sudoku_Game)

**Graph Data Structure**

* The Sudoku board is treated as a graph where each cell is a node, and constraints (rows, columns, and sub grids) form the edges.
* This setup makes it easy to check valid moves since the graph structure enforces Sudoku rules during traversal.

**Breadth-First Search (BFS)**

* BFS explores all possibilities level by level, which ensures every solution is found if there are multiple.
* **Strengths**: Guarantees that it will find all solutions eventually because it systematically checks every possibility.
* **Weaknesses**: Uses a ton of memory since it keeps track of all the nodes at each level. Also pretty slow, especially for complex puzzles.
* Example: For Question 1 in the results, BFS took 9.37 seconds and expanded over 12,000 nodes. Not ideal if speed is critical.

**Depth-Limited Search (DLS)**

* This is basically DFS but with a cap on how deep it can go. If it doesn’t find a solution within the limit, it stops and backtracks.
* **Strengths**: Very memory efficient since it only keeps track of nodes on the current path. Also faster than BFS for simpler puzzles.
* **Weaknesses**: If you set the depth limit too low, it might fail to find a solution entirely.
* For example, in the same test (Question 1), DLS only took 0.61 seconds and expanded just over 1,000 nodes. Big difference from BFS.

**Comparison Between BFS and DLS**

* BFS is better if you care about finding *all* solutions, but it’s slower and memory-hungry.
* DLS is faster and uses less memory but only works well if you set the depth limit correctly.
* In most cases, DLS is more practical for solving standard Sudoku puzzles if all you need is a single solution.

**Performance Results**

**BFS**:

* Question 1: 9.37 seconds, 12,867 nodes expanded.
* Question 2: 2.81 seconds, 4,700 nodes expanded.
* Question 3: In 8.62 seconds, 13,201 nodes expanded.

**DLS**:

* Question 1: 0.61 seconds, 1,188 nodes expanded.
* Question 2: 2.50 seconds, 4,212 nodes expanded.
* Question 3: 1.50 seconds, 2,507 nodes expanded.

**Takeaways**

* BFS is great for puzzles where you need to find multiple solutions or systematically explore all options.
* DLS works better if you’re only looking for one solution and care about speed and memory.
* For our project, combining BFS and DLS makes sense because it gives us the thoroughness of BFS and the efficiency of DLS. Dynamic depth adjustment is key to making that combo work well.