EVALUATION

1. Which algorithm is fastest (finds goal in fewest iterations)?

A: In almost all of the cases, Greedy Best First Search (GBFS) finds goal in fewest iterations and hence is fastest among the three algorithms.

2. Which is most memory efficient (smallest max frontier size)?

A: In most cases, Breadth First Search (BFS) has the smallest max frontier size and hence is the most memory efficient among the three algorithms.

3. Which visits the fewest vertices?

A: Greedy Best First Search (GBFS) outperforms BFS and DFS (Depth First Search) by visiting the fewest vertices in almost all of the cases.

4. Which generates the shortest path length? (Is any of them optimal?)

A: BFS generates the shortest path length among the three algorithms. Note that BFS finds the path with the fewest edges (definition of optimal), though it is not necessarily the shortest path as we are assuming all edges to be of same length. If the edges have the same length in a graph, then BFS will provide the optimal path. On the other hand, in terms of path length, GBFS and DFS are not very optimal. However, GBFS does offer better time complexity than BFS in almost all cases.

5. Are the performance differences what you expected based on the theoretical complexity analysis?

A: Yes, the performance differences are quite similar to the theoretical complexity analysis. In terms of time complexity, it was expected that GBFS would be the one performing better than the other two because we have taken a reasonably good heuristic function. Although DFS and GBFS have the same worst case time complexity of $O(b^m)$ and BFS has time complexity of $O(b^d)$ (considering solution is at depth d for BFS), where b is the branching factor and m is the maximum depth of the tree, GBFS's time complexity can be reduced significantly by using a good heuristic function. This is why GBFS gives us a very good time complexity (lower number of iterations) in our results. BFS and DFS alter in taking up the worst spot depending on BFS's solution being at a greater depth in its solution tree or DFS's maximum depth of the solution tree being higher. In terms of space complexity, BFS has $O(b^d)$ and GBFS has $O(b^m)$ space complexity while DFS has O(bm) space complexity. However, BFS seems to be the most memory efficient from the results. This can be explained considering BFS can find the solution node to be at a much shallower depth and hence a much reduced space complexity. DFS on the other hand, doesn't have an exponential space complexity but still it performs worse considering the high depth of the tree that it may have generated. GBFS performs nearly as well as BFS in most of the cases because of its good heuristic function.

6. Does BFS always find the shortest path? Does GBFS always go "straight" to the goal, or are there cases where it gets side-tracked?

A: BFS finds the shortest path only when the step costs in the path are uniform. GBFS follows a greedy approach and hence it may not always go "straight" to the goal and get side-tracked. For example, GBFS takes a longer path than BFS while traveling from (1, 1) to (20, 20). It goes on towards the solution but due to a blocking object, it gets sidetracked and has to take a longer path. Hence, GBFS may not always go "straight" to the goal.

PERFORMANCE METRICS

Test Case Number	Start	End	Algorithm	Number of Iterations (goal tests)	Maximum Frontier Size	Mean Solution Path Length	Number of Vertices Visited
1	(1, 1)	(20, 20)	BFS	270	21	28	271/275
			DFS	32	36	31	67/275
			GBFS	32	36	31	67/275
2	(1, 1)	(4, 4)	BFS	16	9	3	23/275
			DFS	4	7	3	10/275
			GBFS	4	7	3	10/275
3	(20, 1)	(1, 1)	BFS	122	13	19	133/275
			DFS	161	91	114	245/275
			GBFS	20	20	19	39/275
4	(1, 20)	(20, 20)	BFS	187	21	19	193/275
			DFS	20	20	19	39/275
			GBFS	20	20	19	39/275
5	(2, 3)	(7, 7)	BFS	68	10	9	77/275
			DFS	260	68	9	271/275
			GBFS	10	15	9	24/275
6	(1, 1)	(1, 2)	BFS	2	3	1	4/275
			DFS	271	67	1	271/275
			GBFS	2	3	1	4/275
7	(20, 20)	(1, 1)	BFS	266	23	28	271/275
			DFS	180	86	99	264/275
			GBFS	48	45	31	92/275
8	(13, 6)	(7, 6)	BFS	240	21	28	254/275
			DFS	61	56	58	115/275
			GBFS	41	35	28	75/275
9	(1, 20)	(20, 1)	BFS	271	21	33	271/275
			DFS	39	37	38	75/275
			GBFS	64	51	47	114/275

AVERAGE PERFORMANCE OF TEST CASES

Algorithm	Number of Iterations (goal tests)	Maximum Queue Size	Mean Solution Path Length	Number of Vertices Visited
BFS	160.2222222	15.77777778	18.66666667	166.33/275
DFS	114.2222222	52	41.33333333	150.77/275
GBFS	26.77777778	25.77777778	20.88888889	51.55/275