

### Performance Metrics

Start Point : 1, 1 Goal : 20, 20

Search Type	Iterations	Frontier Size	Vertices Visited	Path Length
BFS	264	<b>21</b>	271	<b>28</b>
DFS	44	43	86	40
GBFS	<b>32</b>	36	<b>67</b>	31

Start Point : 13, 6 Goal : 7, 6

Search Type	Iterations	Frontier Size	Vertices Visited	Path Length
BFS	247	<b>22</b>	263	<b>28</b>
DFS	158	91	232	68
GBFS	<b>40</b>	35	<b>74</b>	<b>28</b>

Start Point : 1, 20 Goal : 20, 1

Search Type	Iterations	Frontier Size	Vertices Visited	Path Length
BFS	270	<b>22</b>	271	<b>33</b>
DFS	212	99	262	46
GBFS	<b>64</b>	51	<b>114</b>	47

Start Point : 9, 11 Goal : 16, 10

Search Type	Iterations	Frontier Size	Vertices Visited	Path Length
BFS	261	23	265	27
DFS	47	42	88	35
GBFS	62	36	97	28

Start Point : 1, 1 Goal : 12, 3

Search Type	Iterations	Frontier Size	Vertices Visited	Path Length
BFS	79	13	91	11
DFS	101	84	183	81
GBFS	12	15	26	11

Start Point : 1, 1 Goal : 4, 4

Search Type	Iterations	Frontier Size	Vertices Visited	Path Length
BFS	16	9	23	3
DFS	4	7	10	3
GBFS	4	7	10	3

## **Evaluation:**

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

-> GBFS is fastest in all the cases except((9,11) to (16,10)) when there is an obstacle in between the start and the goal.

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

-> BFS in all the cases but in case of (1,1) to (4,4), DFS and GBFS proved to be more memory efficient than BFS.

### **3. Which visits the fewest vertices?**

-> GBFS visits fewest vertices in all cases except when there is an obstacle in between the start and the goal.

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

-> BFS always gives the shortest path length but sometimes it GBFS also gives the shortest path but in that case it is equal to BFS .

BFS is optimal only when the path cost is constant and positive while GBFS is not always optimal but it is efficient.

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

-> The differences from theoretical complexity are:

a.) DFS should be most memory efficient but the results show that BFS is more memory efficient in most the cases.

b.) BFS should perform faster than DFS but the result shows that DFS performs faster than BFS.

### **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?**

-> BFS always gives the shortest but sometimes GBFS too gives the shortest path but in that case it is equal to BFS.

If there are obstacles in between the start point and the goal then GBFS gets side-tracked.