

# ARTIFICIAL INTELLIGENCE FOR ROBOTICS

ASSIGNMENT: 04

*Debaraj Barua*

06.05.2017

## Contents

Conclusion	2
------------	---

## Conclusion

From the data given below we can infer that:

1. For Map 1, and Map 2, the DFS performs better as it finds all the dirt earlier. DFS reaches the result navigating a much lower number of nodes in each case, thus takes less time.
2. For the Map 3, the performances of both searches are almost similar. And both searches end up searching for the entire possible node frontier.
3. Since the first two cases show a better case scenario of the depth first search we can conclude that DFS is more effective in these maps. However, if the position of the dirt had varied to a worse case scenario for DFS, it would actually take much more time than the BFS.
4. The space complexity for the DFS search is much better than BFS which is clearly visible from the table below.
5. For the third map, ideally, BFS should take lesser time to complete, however, this is not the case as per data. This may be due to some time calculation, as I am using `std::clock()` to calculate time required, and this freezes time for certain multi threading sleep operations. As such, the time required calculated via the process may be actually not very accurate in this case.

	Time (ms)		Visited Nodes		All Goals	
	BFS	DFS	BFS	DFS	BFS	DFS
<b>MAP 1</b>	447	384	2322	1400	Yes	Yes
<b>MAP 2</b>	370.75	346.74	1893	981	Yes	Yes
<b>MAP 3</b>	953	913	4618	4618	No	No

Figure 1: Conclusion of BFS and DFS implementation