Search Methods Coursework Assignment

FAQ

Richard Watson

1) Aims. Some students interpret this assignment as a challenge – to create an algorithm that finds a solution to the problem. But note that the assignment asks you to investigate how the ability of various algorithms scales-up with problem difficulty. Some of the algs don’t scale well, some do – if you ‘fix’ them all so they all work well then you aren’t answering the question about how well (or not) the algorithms scale. In other words, implement the algorithms you are asked to implement, and show how well they scale or how they fail. THEN, if you’re looking for the extra marks under ‘Evidence of creativity/additional technical challenge/ambition’ go ahead and modify the algs in ways you think are sensible.

In particular, you are asked to implement depth first search. Depth first search has obvious problems on this problem. I mentioned in class that you might like to randomise the order in which actions are selected so that there’s a chance of getting out of pointless looping behaviours. But, if you keep a list of previously visited nodes, then (although that’s a sensible thing to do to find solutions) that changes the algorithm significantly. Specifically, if you do that then you’ve implemented GRAPH SEARCH not TREE SEARCH. There is a good chance that graph search will work better than tree search – but graph search has properties that are importantly different to DF tree search. In particular, consider the space requirements. The point of using DFS and likewise IDS is to get that polynomial space complexity – you don’t have that if you store every node visited!

In sum, the aim of the assignment is for you to understand the strengths and weakness of the standard algorithms. Show me that you understand that. Then if you want to try and improve on them – (eg w graph search) that’s a bonus.

I think my colleague put it well: “The majority take [the aim of the assignment] as ‘most of these search methods suck and won’t be able to solve the puzzle, so your job is to fix it’, and the minority take it as the assignment it truly is: ‘understand the distinction between these search methods, show me that you implemented them, and confirm the expected time complexities’ (anything else is an extra).”

2) The assignment doesn’t specify exactly how to control problem difficulty. But it does suggest that controlling the depth of the least cost solution is a good way to do it.

3) The assignment doesn’t ask you to plot results of space complexity (only time) – but that’s a perfectly reasonable thing to do in ‘evidence of creativity...’. (Say something about the conclusions you draw rather than merely plotting the data)

4) The goal state is exactly the one shown (except that the agent position doesn’t matter). And the start state is exactly the one shown. But if you’ve done something different, say what you’ve done in the report.

5) Make it easy for me to give you credit for the things that you have done. Use the headings specified – these point out the items in the marks scheme. In particular, if you think you’ve done something that goes beyond the basics – put it under extras (rather than burying it in the details).
6) There is a 6 page limit. Try to keep the writing and the main results figures in that limit. But this is a guide to let you know how much writing is expected. The example output (and code) need not be included in the limit.

7) Do the time complexity plots on one plot not separate plots for each algorithm (otherwise it’s difficult to see the comparison between them clearly – especially if you don’t take care with using the same scale on the y axis in all plots).

8) The time I ask you to plot is NOT CPU time or clock time! The instructions on this point are clear.

9) Make sure you provide evidence that the algorithms have been implemented correctly (not just the time results).

10) It might be the case that some (modified) algs have better time complexity that A*. But that’s not (necessarily) the whole story… is it. Show me that you understand the trade-offs.