1) The way DFS works is by traveling down the graph from the top to the bottom looking at all paths starting at the left and moving rightwards (if you represent the graph as a tree). This process continues until we find a solution. This means that we can certainly find a solution without exploring all nodes. This makes it not surprising that we explore the closer nodes many times, as they must be traveled for many lower paths. Also not surprising, there are many nodes we don't visit because the DFS algorithm finds the solution before exploring them.

2) Both A* and UCS have a score of 300, but A* explores 539 nodes while UCS explores 620 nodes. This makes sense because, since the openMaze has few walls, the heuristic should very accurately compute the distance to the goal state. This allows the A* algorithm find the ideal path much faster than the uninformed UCS. Regardless of calculation speed, both algorithms are able to find the optimal solution so of course the two should end up with the same score.

3) I think a good example of needing to find a solution and dealing with the optimality-speed trade-off is mapping. If you are planning your trip somewhere before you go, you have a few minutes. You can let the algorithm run and will be happy if the program returns the fastest journey. On the other hand, if you are driving somewhere and miss a turn, you need the answer fast. The algorithm might not find the fastest way there, but it's more important to have ANY way to where you are going while you are driving. Driving on the road and not knowing where you are going is not a desired feeling.