Lab 2 TDDC17

Oskar Håstad, Martin Hinnerson

Version 0.1

2018-10-25

Status

Reviewed		
Approved		

Project identity

2018/Autumn term
Tekniska högskolan vid Linköpings universitet, IDA

Name	Telephone	Email
Oskar Håstad		oskha905@student.liu.se
Martin Hinnerson	072 210 43 72	marhi386@student.liu.se

Course responsible: Patrick Doherty, patric.doherty@liu.se
Lab Assistant: Axel Strandberg, axest863@student.liu.se

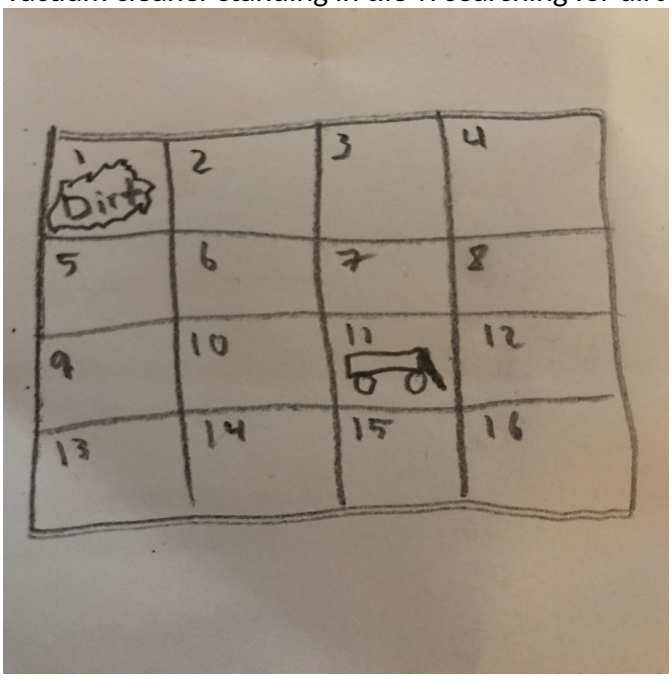
1 Part 1

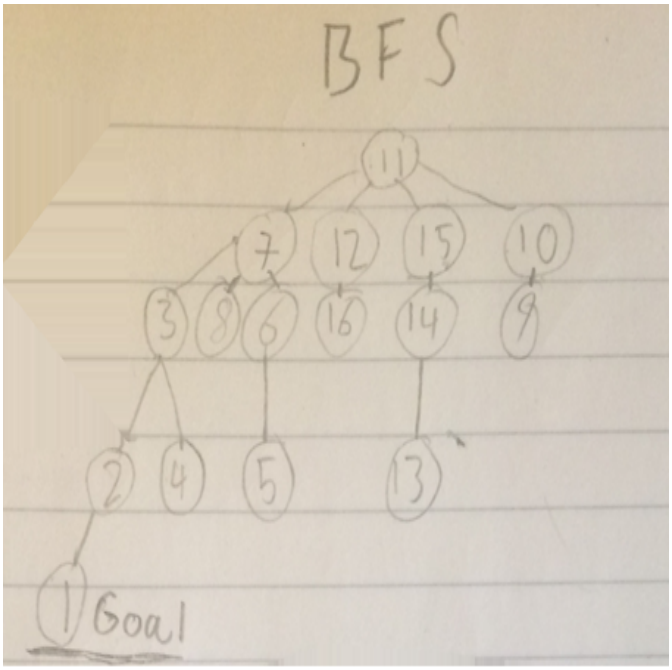
Part one can be seen in the code.

2 Part 2

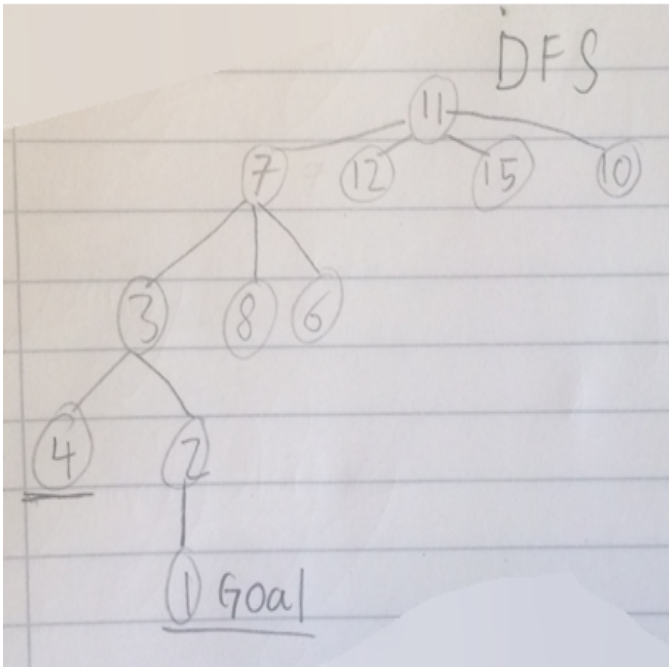
Here are the answers to the questions in part 2 of the lab:

1. Branching factor is 4. The different states of the agent could be, start state, goal state, and the states in between which could be, dirt found, wall found. The actions used by the agent are move right, move left, move up, move down and suck dirt.
2. There is no difference between Breadth First Search and Uniform Cost Search if the cost of all actions are 1.
3. a) $\frac{h1+h2}{2}$ is admissible because the average of $h1$ and $h2$ can't be greater than the biggest one of them.
   b) $2h1$ is not necessarily admissible.
   c) $max(h1, h2)$ is obviously admissible if both $h1$ and $h2$ are admissible.
4. An admissible heuristic could be the shortest path (euclidean distance) from the current node to the goal node, i.e. $h(n) = (\Delta x + \Delta y)$
   The cost function $g(n)$ could be the distance travelled. The euclidean distance is an admissible heuristic because it is impossible for the vacuum cleaner to find a shorter path than this.
5. Vacuum cleaner standing in tile 11 searching for dirt in tile 1.



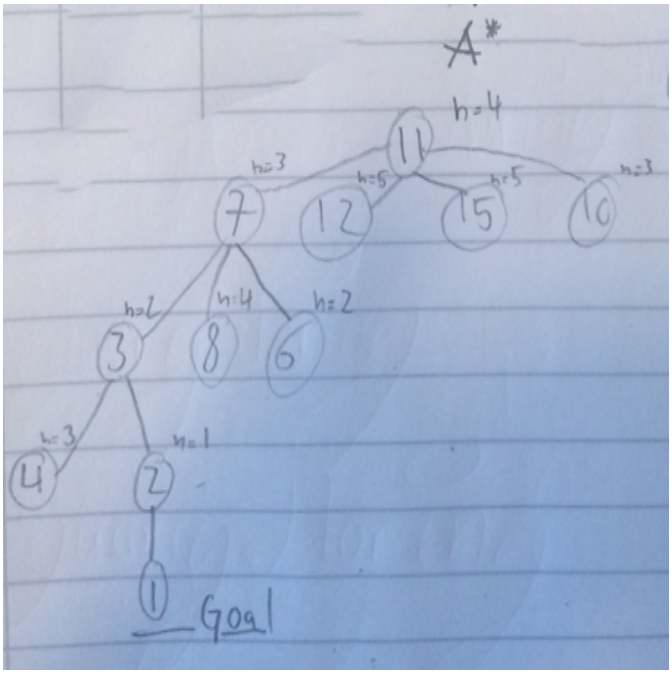


$MemoryUsage : O(branchingFactor^{depth}), BranchingFactor = 4, Depth = 2.$



$MemoryUsage : O(branchingFactor^{depth}), BranchingFactor = 4, Depth = 2.$

A* using the Manhattan-distance as heuristic



$MemoryUsage : O(branchingFactor^{depth}), BranchingFactor = 4, Depth = 2$.

6. **Breadth First Search** is complete and optimal for unit step costs. It is complete because it can backtrack an thus never get stuck in a node.
   Uniform Cost Search is optimal for a general step cost. It can however get stuck in a infinite loop if there is a path with an infinite sequence of zero cost actions.
   Depth First Search is neither optimal or complete. It is not optimal because it can expand down in the tree and finish when a goal is found even if it is not the solution with the lower cost. For the same reason as UCS this algorithm is not complete.
   A* Search is both complete and optimal. **Depth Limited Search** is not complete if the goal is deeper than the limit because then it will never find the solution. For the same reason it is not optimal.
   Iterative Deepening Depth First Search This is a version DLS and it is complete if the branching factor is finite and it is optimal if each step cost is the same.
   Bidirectional Search is optimal and complete if BFS is used on both sides with unit step cost.
   A* Search is optimal if the heuristic function is admissible and if it is consistent. It is also complete in finite spaces.

7. In Lab 1/Task 2 we used breadth first search. The result was pretty good and it would be hard to make better with another search algorithm. A* search would not be possible because it needs a heuristic function and we don't know anything about the environment in the beginning. It could be used to find home after everything is explored.