TDT4136 - assignment 3

Aleksander Hansen

October 2015

A.1.2

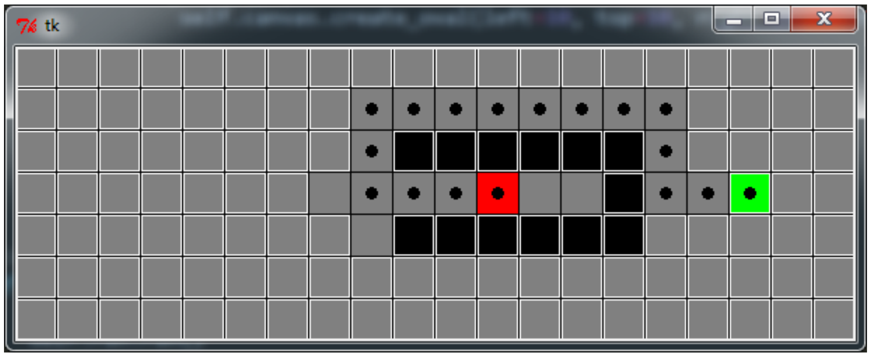

Figure 1: Board: board-1-1.txt, Algorithm: A*

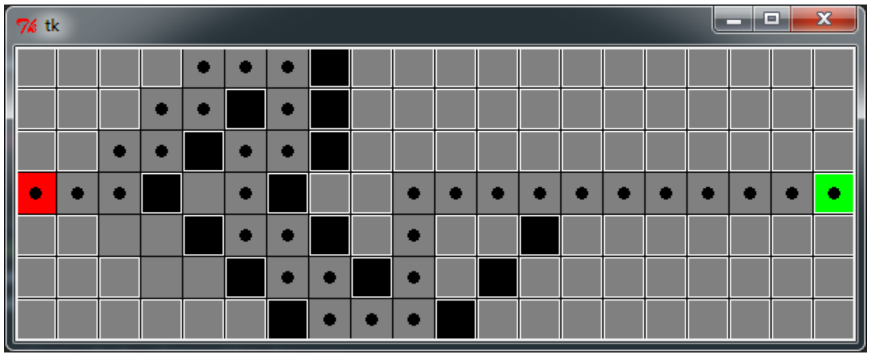

Figure 2: Board: board-1-2.txt, Algorithm: A*

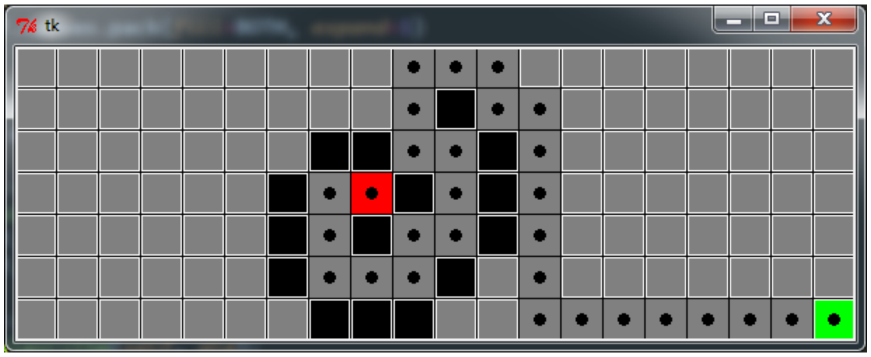

Figure 3: Board: board-1-3.txt, Algorithm: A*

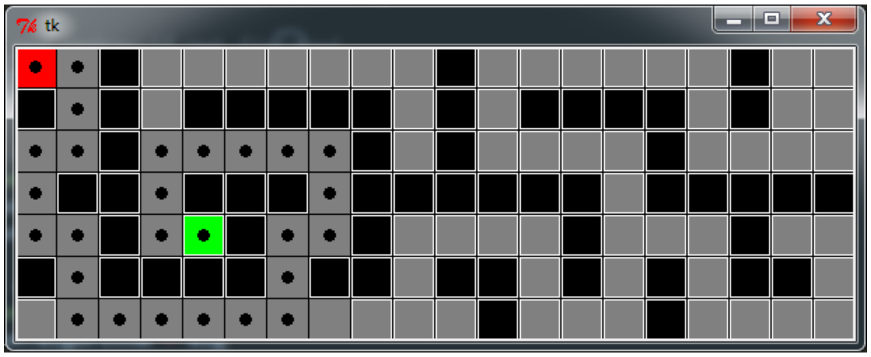

Figure 4: Board: board-1-4.txt, Algorithm: A*

A.2.2

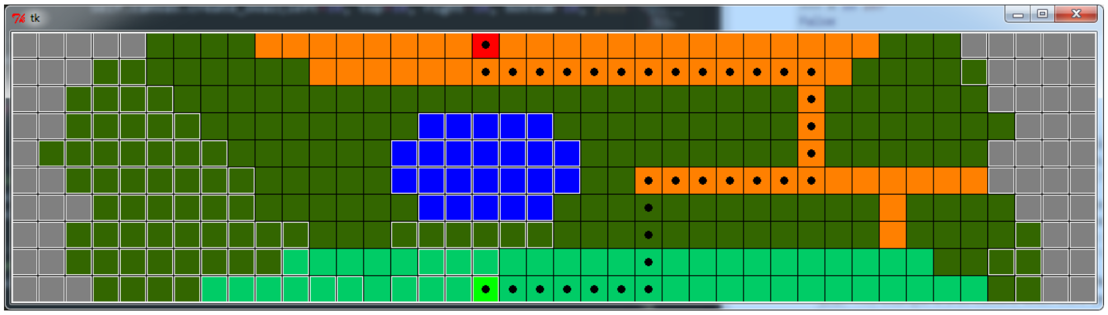

Figure 5: Board: board-2-1.txt, Algorithm: A*

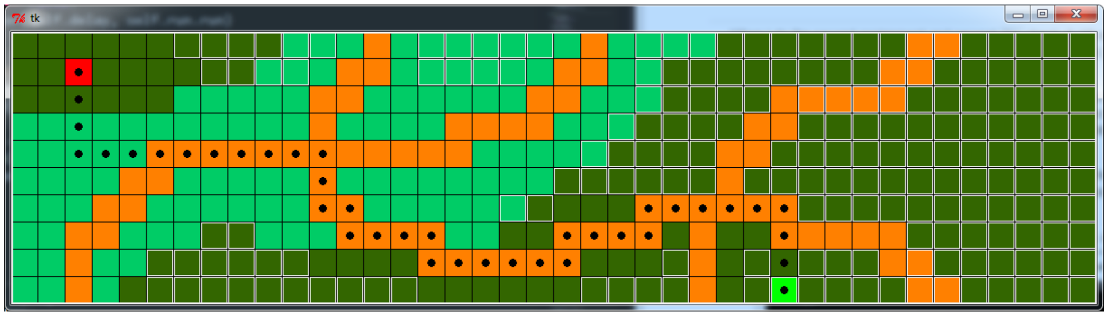

Figure 6: Board: board-2-2.txt, Algorithm: A*

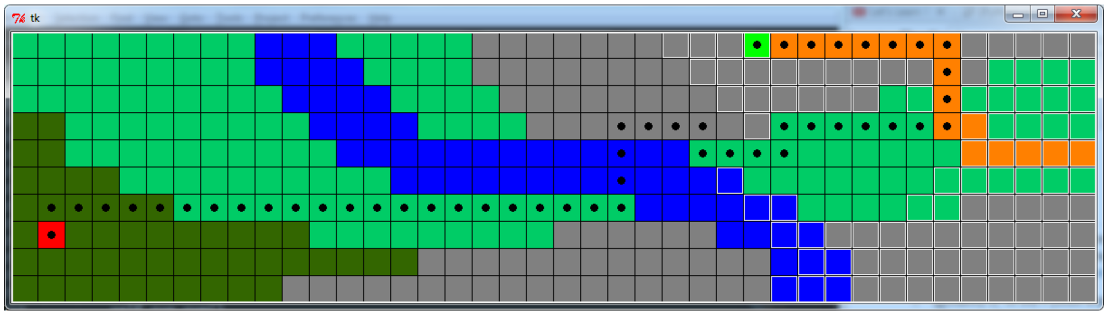

Figure 7: Board: board-2-3.txt, Algorithm: A*

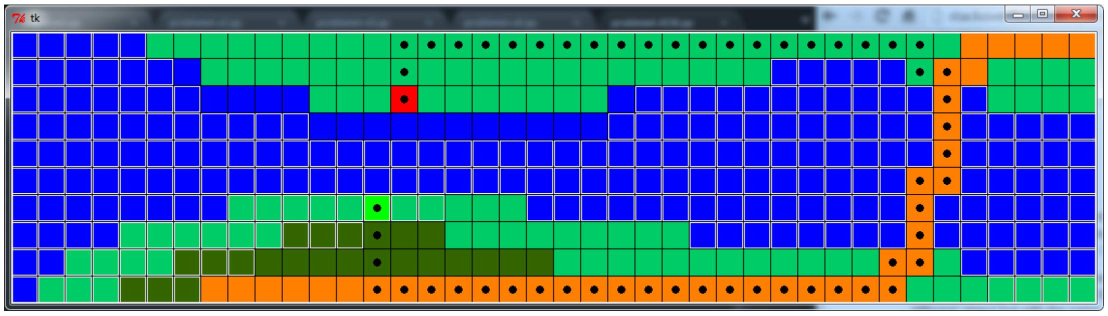

Figure 8: Board: board-2-4.txt, Algorithm: A*

A.3.2

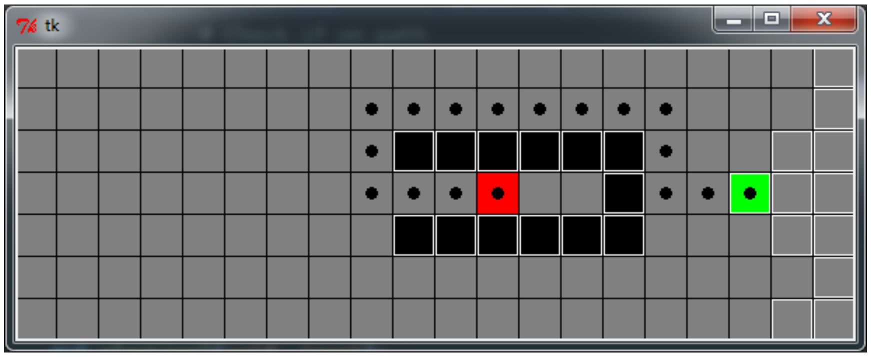

Figure 9: Board: board-1-1.txt, Algorithm: BFS

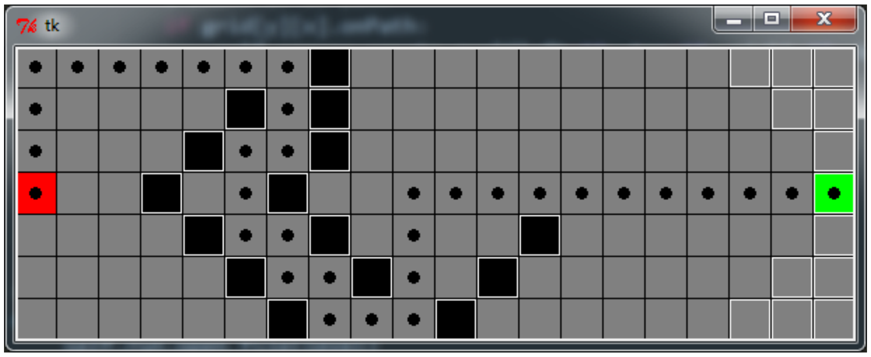

Figure 10: Board: board-1-2.txt, Algorithm: BFS

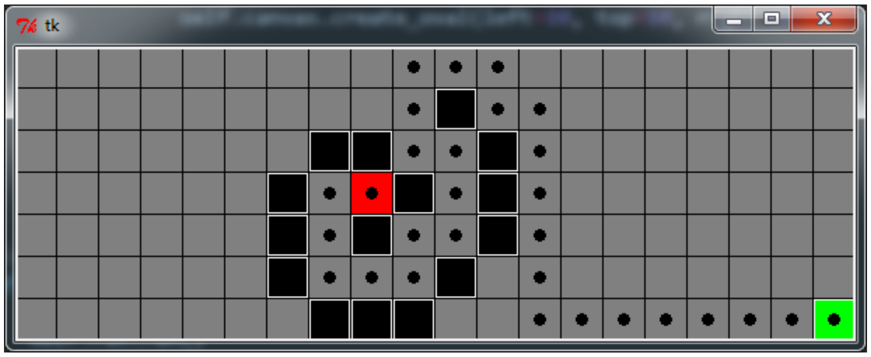

Figure 11: Board: board-1-3.txt, Algorithm: BFS

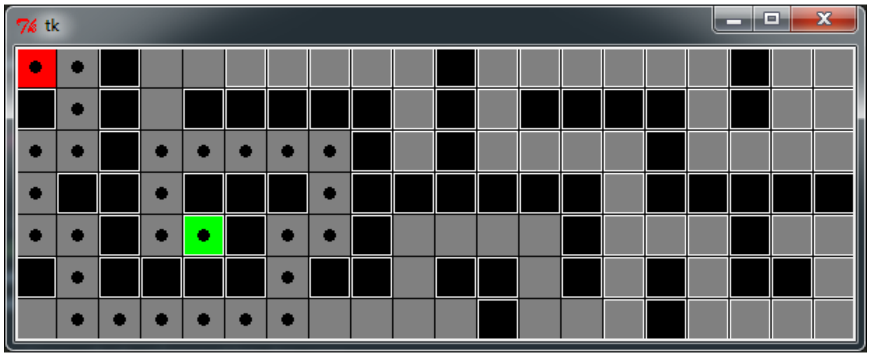

Figure 12: Board: board-1-4.txt, Algorithm: BFS

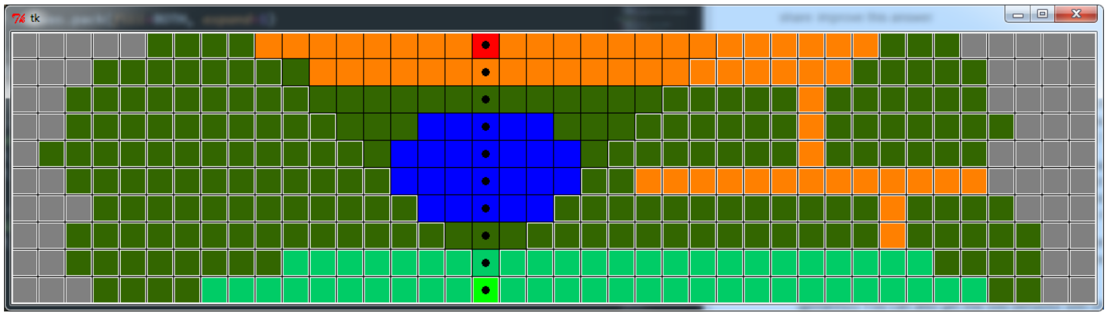

Figure 13: Board: board-2-1.txt, Algorithm: BFS

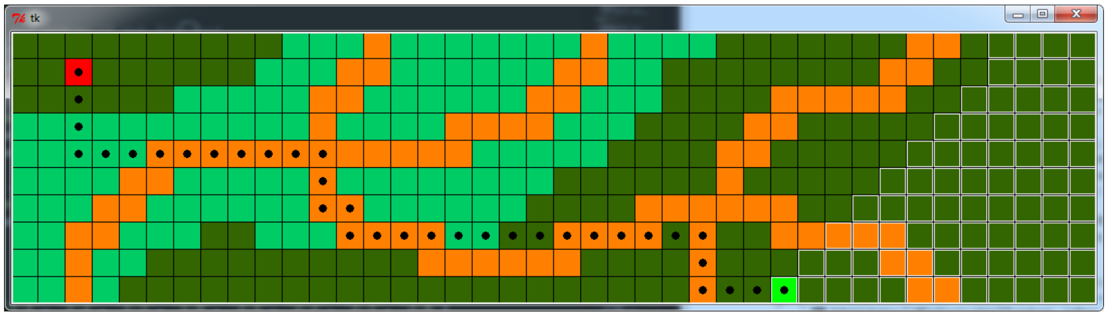

Figure 14: Board: board-2-2.txt, Algorithm: BFS

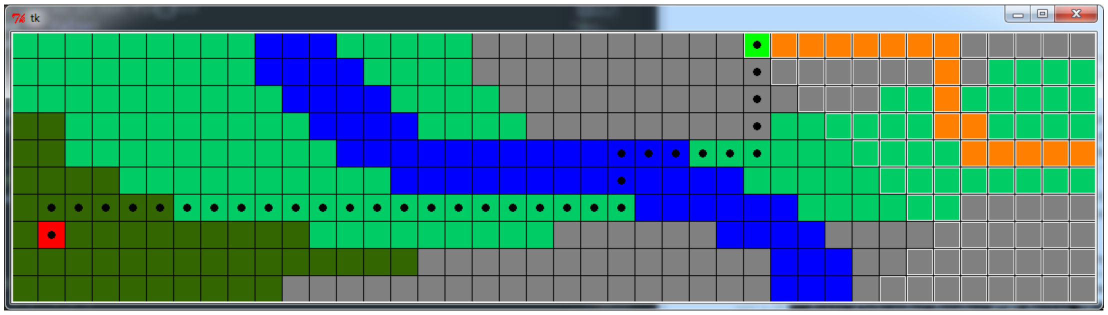
Figure 15: Board: board-2-3.txt, Algorithm: BFS

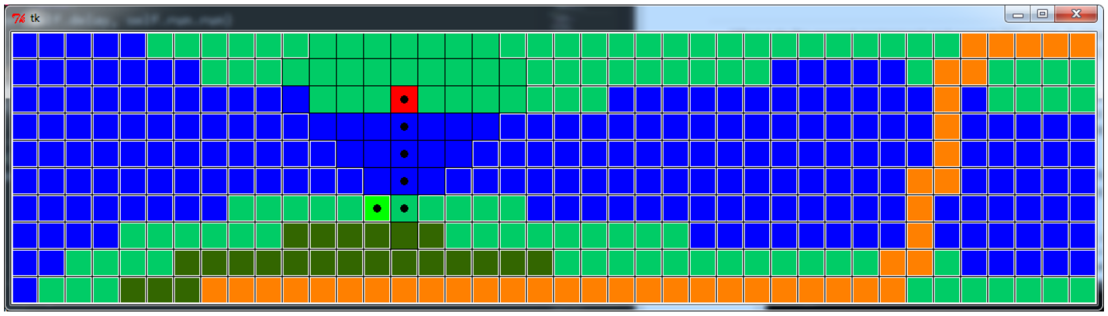
Figure 16: Board: board-2-4.txt, Algorithm: BFS

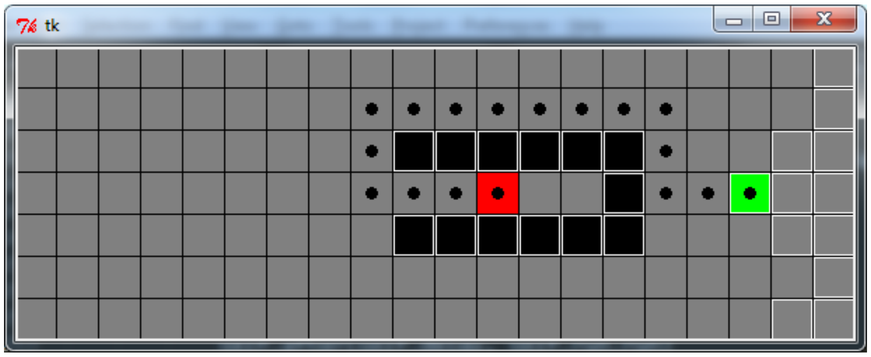
Figure 17: Board: board-1-1.txt, Algorithm: Dijkstra

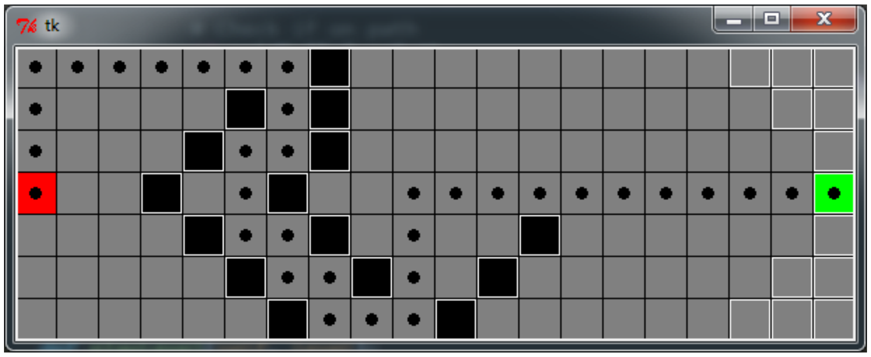
Figure 18: Board: board-1-2.txt, Algorithm: Dijkstra

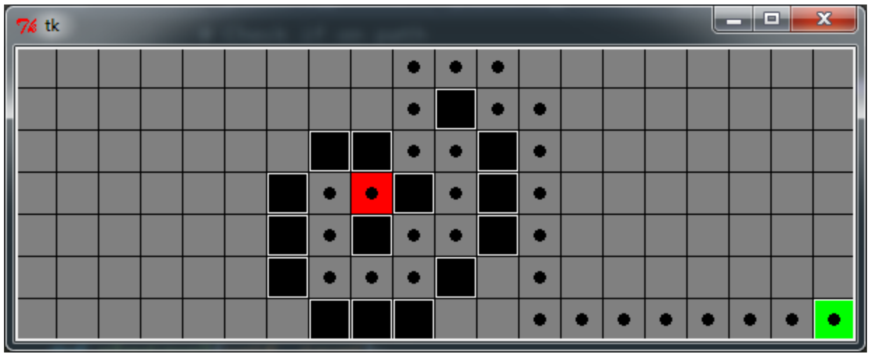
Figure 19: Board: board-1-3.txt, Algorithm: Dijkstra

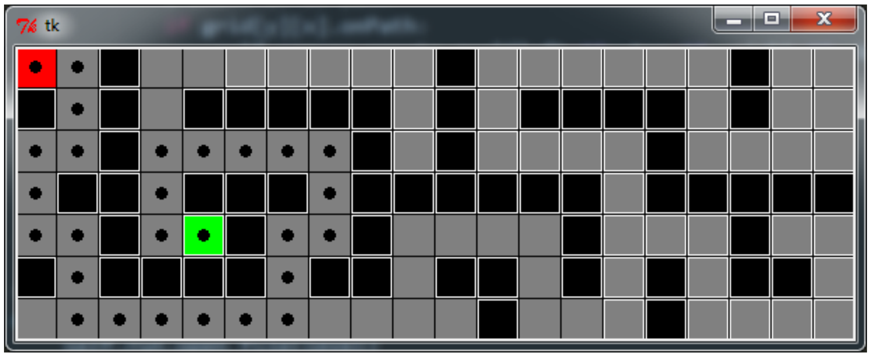
Figure 20: Board: board-1-4.txt, Algorithm: Dijkstra

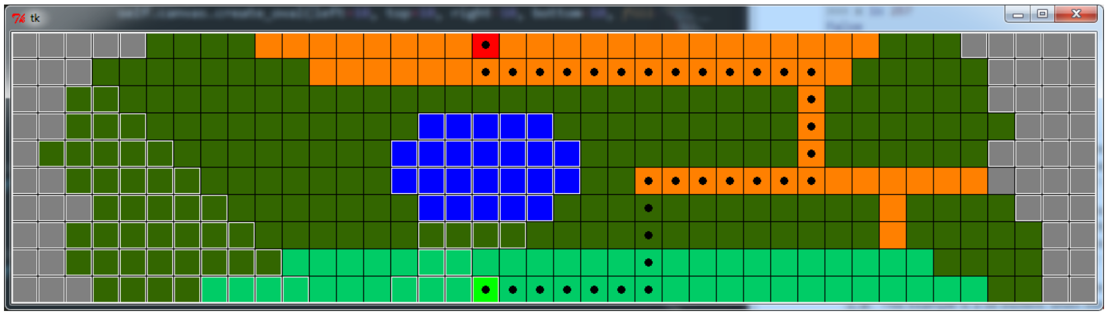

Figure 21: Board: board-2-1.txt, Algorithm: Dijkstra

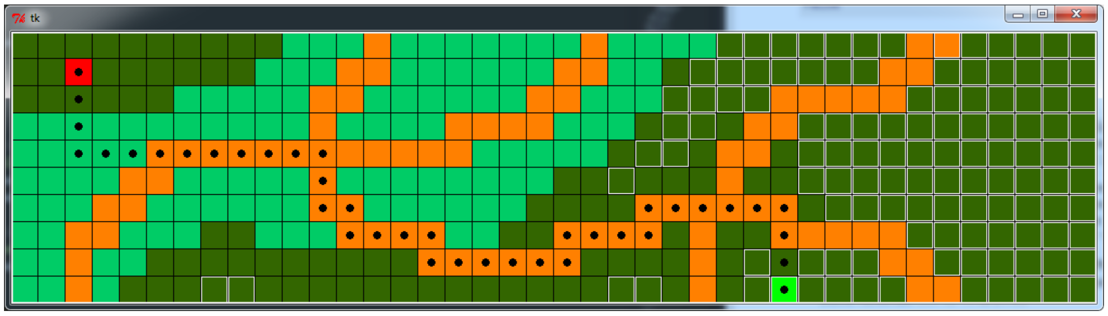

Figure 22: Board: board-2-2.txt, Algorithm: Dijkstra

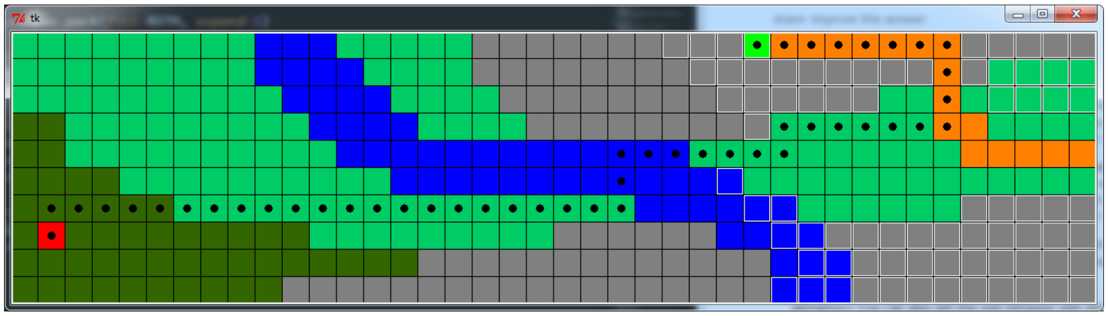

Figure 23: Board: board-2-3.txt, Algorithm: Dijkstra

A.3.3

In board-1-1 we see that all the algorithms find the same path, but A* opens and closes way fewer nodes than BFS and Dijkstra.

In board-1-2 A* takes a different path than BFS and Dijkstra takes, but the cost is the same. Again Dijkstra and BFS opens and closes the same amount of nodes.

In board-1-3 and board-1-4 we have the same conclusion as in board-1-1.

In board-2-1 BFS takes the naive path straight over the water, ignoring the cost, but opens the least amount of nodes. A* and Dijkstra finds the same path, but A* does it by opening fewer nodes.

In board-2-2 A* and Dijkstra finds the same path again, but A* does it faster. BFS unsurprisingly finds a sub optimal path and is the slowest.

In board-2-3 A* finds the most cost-effective path again and is fastest. Dijkstra finds an equal cost-effective path, but is slower. BFS opens the most nodes and is therefore the slowst. it also cuts over the mountain rather than going the long eaiser way around.

In board-2-4 BFS finds the shortest path, but most expensive path of the three, but does it quickest. A* and Dijkstra finds the same path, but again A* does it faster.