

Greedy Best First search

We add attributes to the Node object. We add a heuristic and the path cost, to reach that node. We modify the REMOVE_FIRST function, such that the node which has the lowest heuristic value is removed. In this way we always choose the Node to expand, which has the lowest heuristic value. Solution path of this method:

```
State: K - Depth: 3 - Heuristic: 0
State: H - Depth: 2 - Heuristic: 1
State: D - Depth: 1 - Heuristic: 2
State: A - Depth: 0 - Heuristic: 100
```

A* Search

With the A* search we need to combine the heuristic and the path cost into the evaluation function.

```
State: L - Depth: 4- Heuristic: 0
State: H - Depth: 3- Heuristic: 1
State: E - Depth: 2- Heuristic: 4
State: B - Depth: 1- Heuristic: 5
State: A - Depth: 0- Heuristic: 100
```

Weighted A*

We choose to bias in favour of the heuristic (it estimates the cost of reaching goal from node), as it is admissible (it does not overestimate the cost of reaching the goal).