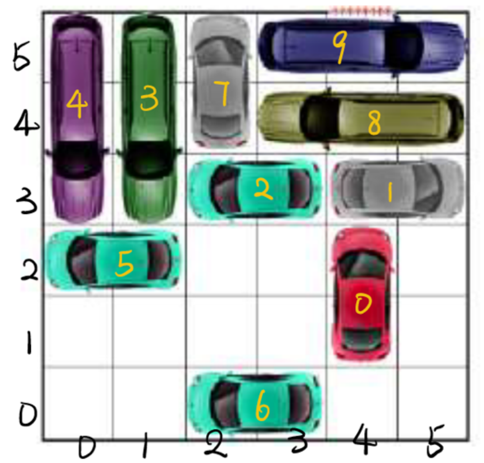
Lab 1
Report

1. State Space: Given the initial state, the state space contains all possible arrangements of cars and trucks resulted from the valid movements from the initial state. The horizontal vehicles will always be at the initial row, vertical vehicles are always at the same column, and the red car is always aligned with the door.
2. Initial State: The given board with the given position of each vehicle.
3. Action: move one vehicle such that it does not collides with another vehicle or exceed the board limit, forward or backward for one square.
4. Transition model: Input: The current board with the current arrangement of vehicles + Action: Move one valid vehicle forward or backward for one square ➔ Output: The board all other vehicles keep the same, except for one vehicle has been moved. Step cost is 1 for each move.
5. Goal Test: The red car touches the door.
6. Lower and upper bound on branching factor: The lower bound is 0 which means no vehicles can move. The upper bound is 2n, which n is the number of the vehicles, this means all vehicles can move in both of their sides.
7. Lower and upper bound on the solution depth of this solution: The lower bound is 0 which means the red car has already touched the door. The upper bound is 5^{n} , where n represents the number of the vehicles in total. Since for each car, there are at most 5 moves, thus for n vehicles, there can be at most 5^{n} move to the solution.
8. Optimal Solutions
   Assumption: All the arrangement keeps the same except for the moved one; the notation is based on the index of the vehicle; the following graph shows each index of the vehicles.

   a. Initial state A: Cost 14
      i. Initial State
      ii. Vehicle 2 move up
      iii. Vehicle 1 move left
      iv. Vehicle 1 move left
      v. Vehicle 2 move up
      vi. Vehicle 3 move left
      vii. Vehicle 5 move down
      viii. Vehicle 5 move down
      ix. Vehicle 5 move down
      x. Vehicle 4 move left
      xi. Vehicle 0 move up
      xii. Vehicle 6 move left
      xiii. Vehicle 7move left
      xiv. Vehicle 0 move up
      xv. Vehicle 0 move up

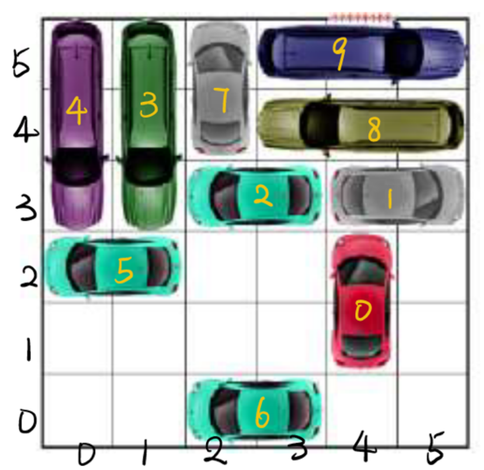
Initial State A

   b. Initial State B: Cost 54
      i. InitialState
      ii. Vehicle 5 move right



Initial State B

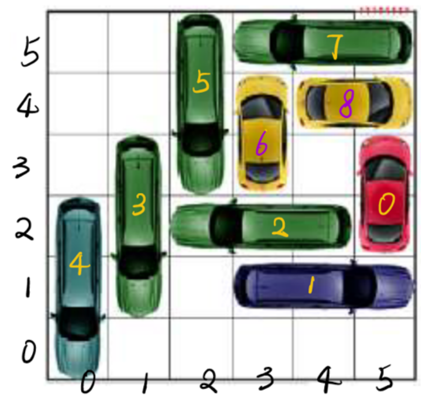
iii. Vehicle 5 move right
iv. Vehicle 3move down
v. Vehicle 3move down
vi. Vehicle 3move down
vii. Vehicle 4move down
viii. Vehicle 4move down
ix. Vehicle 0move down
x. Vehicle 4move down
xi. Vehicle 2move left
xii. Vehicle 2move left
xiii. Vehicle 7move down
xiv. Vehicle 5 move right
xv. Vehicle 7move down
xvi. Vehicle 7move down
xvii. Vehicle 2 move right
xviii. Vehicle 4 move up
xix. Vehicle 2 move right
xx. Vehicle 3 move up
xxi. Vehicle 6 move left
xxii. Vehicle 6 move left
xxiii. Vehicle 7 move down
xxiv. Vehicle 5 move left
xxv. Vehicle 3 move up
xxvi. Vehicle 3 move up
xxvii. Vehicle 4 move up
xxviii. Vehicle 5 move left
xxix. Vehicle 4 move up
xxx. Vehicle 5 move left



Initial State B

xxxi. Vehicle 7 move up
xxxii. Vehicle 6 move right
xxxiii. Vehicle 6 move right
xxxiv. Vehicle 0 move up
xxxv. Vehicle 6 move right
xxxvi. Vehicle 7 move down
xxxvii. Vehicle 5 move right
xxxviii. Vehicle 5 move right
xxxix. Vehicle 3 move down
xl. Vehicle 3 move down
xli. Vehicle 4 move down
xlii. Vehicle 3 move down
xliii. Vehicle 2 move left
xliv. Vehicle 1 move left
xlv. Vehicle 4 move down
xlvi. Vehicle 4 move down
xlvii. Vehicle 2 move left
xlviii. Vehicle 9 move left
xlix. Vehicle 1 move left
l. Vehicle 9 move left
li. Vehicle 0 move up
lii. Vehicle 8 move left
liii. Vehicle 8 move left
liv. Vehicle 0 move up
lv. Vehicle 0 move up

c. Initial State C: Cost 21
   i. Initial State
   ii. Vehicle 3 move up
   iii. Vehicle 3 move up
   iv. Vehicle 2 move left
   v. Vehicle 4 move up
   vi. Vehicle 4 move up
   vii. Vehicle 1 move left
   viii. Vehicle 1 move left
   ix. Vehicle 1 move left
   x. Vehicle 4 move up
   xi. Vehicle 2 move left
   xii. Vehicle 6 move down
   xiii. Vehicle 6 move down
   xiv. Vehicle 8 move left
   xv. Vehicle 0 move up
   xvi. Vehicle 6 move down
   xvii. Vehicle 2 move right
   xviii. Vehicle 2 move right
   xix. Vehicle 2 move right
   xx. Vehicle 5 move down
   xxi. Vehicle 7 move left
   xxii. Vehicle 0 move up



Initial State C

9. Performance Analysis

In this program we implemented two heuristic functions h1 and h2.
For current state A:
The heuristic cost function h1 calculates how many vehicles blocking the red car from the door.
The heuristic cost function h2 calculates the sum of how many vehicles blocking the red car from the door, added with how many vehicles blocking these vehicles from moving away, notice that we only consider the number of cars that directly block these cars that block the red car directly.
Proof for admissible:

1. Suppose there are n vehicles directly blocking the red car from the door a. Then the current state needs to take at least n steps to move the n cars away from the current column such that the path between the red car and the door is cleared. Therefore, h1 is admissible.
2. Suppose there are n vehicles blocking the red car from the door. In addition, there are x vehicles blocking these n vehicles from moving away. Therefore, in order to drive the red car to the door, the current states need at least n moves to remove the vehicles from the current column. In order to move these vehicles from the current column, the current needs at least x moves to remove the blocking cars, thus leaving room for the n cars to move away. Therefore, h2 cost x+n is admissible.

Compare number of states explored:

Explored States\ h functions	H1	H2
Initial State A	850	823
Initial State B	2397	2399
Initial State C	363	352

Reason for similarities and differences:
The numbers of explored states for initial state B are almost the same, while there is only slight difference between initial h1 and h2 for State A and State C. The reason for the similarity in state B maybe that, the vehicles in the initial state are more clustered than those in A and C. Therefore, counting the additional vehicles blocking those in the same column of the red car does not make a big difference.

The reason for the difference for state A and C can be when the vehicles are less clustered, the difference of h2 between each state can be relatively obvious compared to h1. In the more clustered case, counting the nearest vehicles that block those at the same column of the red car is not relatively important, since moving the red car also depends on the number of other vehicles in the cluster if they stuck together. However, in the less clustered case, the mobility of the red car is less dependent on vehicles that are not calculated in the heuristic function, thus the h2 function makes less clustered case more optimal. In other words, when the cars are less clustered, the difference of h2 between each state can be relatively more obvious which can prioritize some of the states.