Home Work 1

Due Date: Not sooner than September 30th. It will be defined next week.

Exercise 1:

Rush Hour Jr. is a board game for kids 5 and up, in which an ice-cream truck has to get out of a traffic jam (Amazon ASIN B00GRV5JNY). The board is a grid of 6 by 6 and the ice-cream truck is always located in some place on the third row and the exit is on the right hand side. The goal is to bring the ice-cream truck out of the grid. The other cars and buses are located on the grid to impede the ice-cream truck circulation and they need to be moved to allow the ice-cream truck to advance. The cars and buses are either on a row or a column and they can only move forward or backward in that row or column. The cars measure 2 grid spaces and the buses 3. The actions allowed are only to move forward or backward by one element and the cost of moving one position is 1.

The input to the program is a list of n-tuples, one per vehicle, each with 4 characters: VXYZ.

- V indicates the type: (C)ar, (B)uses or (I)ce-cream truck).
- X indicates the direction: (H)orizontal or (V)ertical.
- Y indicates the row: A . . . F.
- Z indicated the column: 1 . . . 6.

If the vehicle is placed vertically, the row is the top row occupied by the vehicle and if the vehicle is placed horizontally, the column if the leftmost column occupied by the vehicle.

The first vehicle in the list is always the ice-cream truck (that measures 2 units as the cars). For Image 1 the input would be: IHC3, CVA3, BHB4, CVC5, CVE5.

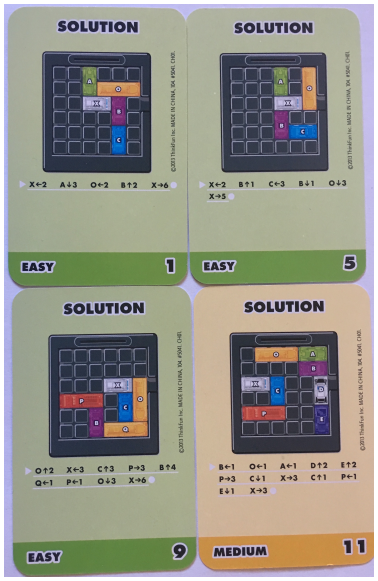
You need to:

a) Implement the iterative-deepening depth-first search.
b) Implement SMA* algorithm using the following heuristic: number of steps the ice-cream truck need to take to get to the output plus the number of vehicles blocking row C that needs to move to make room for the ice-cream truck.
c) Propose a better heuristic for A* start algorithm and discuss why.
d) Discuss what would happen (tree structure, depth, previous heuristic . . .) if the action is changed to moving one vehicle any number of grid elements and the cost is changed to per vehicle movement independent of the number of advanced grid units.

You need to hand-in:

1) The code (any language).
2) The tree for the limit 3 iteration for Easy 1 (You can use the same description used for the input to represent each state). [1 point]
3) The depth of the tree for Easy 1 and 5 and the number of visited nodes. [2 points]
4) The number of visited nodes for SMA* algorithm for all images and the effective branching factor. [3 points]
5) The new heuristic and its discussion. [0.5 points]
6) The discussion for d). [0.5 points]

(a)



(b)

Figure 1: Images for Exercise 1.

Exercise 2:

SoDuKu is a constraint satisfaction problem that can be solved by inference. Code the Belief Propagation algorithm to solve a SuDoKu. The code has to be able to take as input a 9 by 9 matrix representing the SuDoKu and return the solution when the BP algorithms stop iterating. The BP might not converge to the final solution and some squares might take multiple values.

Discuss a way in which the code can be improved to solve SuDoKus with less known values.

You need to hand-in:

1) The code (any language).
2) The solution for SuDoKu in the image when using your code. [2 points].
3) The messages that variable A1 sends to its three factors and the message that the factors send to A1 in each iteration. [0.5 points]
4) The discussion. [0.5 points]

	1	2	3	4	5	6	7	8	9
A			3		2		6		
B	9			3		5			1
C			1	8		6	4		
D			8	1		2	9		
E	7								8
F			6	7		8	2		
G			2	6		9	5		
H	8			2		3			9
I			5		1		3		

Figure 2: Image for Exercise 2.