# FIT 3080

# Assignment 1part 1 report

Takayuki Kimura

28575393

In this assignment, I implemented both DLS and A* in graph search procedure.

## 1) Justification for heuristic function for algorithm A*

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

My heuristic function is using Euclidean distance to calculate h value. The reason why I apply this method is that, as compared to Manhattan distance, Euclidean distance considers of diagonal move between the point A and point B. Since the ROBBIE can move diagonally, it is better to use Euclidean distance to calculate heuristic value. One extra feature is that I divide the coordinates by 2 because the cost that takes to move diagonally is half of the cost to move horizontally and vertically, thus the Euclidean distance assumes all costs are same, however, in this situation it is not same. So the actual Euclidean distance should be half as if it moves diagonally, the cost is half. If the distance, the line between point A and point B is straight line it should not divide it by 2 checking if both points are in same row or same column.

```
if X1 ==X2 or Y1 == Y2:
    S = (X1,Y1)
    G = (X2, Y2)
else:
    S = (X1/2,Y1/2)
    G = (X2/2, Y2/2)
distance = math.sqrt(sum([(m - n) ** 2 for m, n in zip(S, G)]))
```

## 2) Tie breaking rule

My tie breaking rule in A* is retry the comparison based on value g instead of value f because value g is the actual value that has been added from the start node to the node, thus it is more concrete value than assumed value h which was my second option for tie breaking rule.

## 3) Admissible or monotonic

The function for A* is admissible which is one that never overestimates the cost to reach the goal  h(n) <= h*(n). The value h(n) is the heuristic value and Euclidean distance at least returns the value equal to or less than the actual cost.

For monotonicity, as below image shows the equation, the heuristic value must be less than or equal to the cost of getting to n' by action a + the estimated cost of reaching the goal node from n'.

$$h(n) \leq c(n, a, n') + h(n') \; .$$

As far as I can see, the function in my code is also monotonic because it always guarantees that estimated value, h(n) is the shortest distance out of all possible path, thus it is not possible that it could be greater than the c(n,a,n') + h(n'). Hence, it can be said that this algorithm is also monotonic.

## 4) Is it A or A*

If the algorithm is A*, it meets the requirement that heuristic value is always less than or equal to the actual cost of optimal (minimum) path which is h*(n)

h(n) <= h*(n).

Since my algorithm heuristic value is h(n) <=h*(n), it is A* search algorithm.

## 5) Screenshots of test maps and the output

## And analysis of the two algorithms for all test maps

Input1:

```
3
SRR
RRX
RRG
```

DLS output is

```
S-RD-D-R-G 5
0.1178138256072998 the run time
```

A* output is

```
S-RD-D-R-G 5
0.11045551300048828 the run time
```

In the input1, both DLS and A* outputs are optimal and there are no large difference between DLS and A* in terms of the run time as the map in input1 is quite simple. However, the A* visited S-RD-LD before it visits actual path, S-RD-D because my heuristic function tends to assume diagonal move is much lighter than other moves which might have affected on the run time.

Input2:

```
5
SRRXG
RXRXR
RRRXR
XRXRR
RRRRX
```

DLS output is

```
---------------------------------
S-D-D-R-D-D-R-R-U-R-U-U-U-G 24
0.4172093868255615 the run time
```

A* output is

```
---------------------------------
S-D-D-R-D-D-R-R-U-R-U-U-U-G 24
0.36171650886535645 the run time
```

Both outputs are optimal. DLS and A* has very close run time as both mostly does not go into incorrect path often during its execution.

One problem that happened in A* is that since S-R and S-R-R are closer to the Goal position, Euclidean distance will be smaller than others though it is not optimal path. Thus it can be said there is an issue like this problem.

```
-------------------------------------------
N1 S-R 3 2 3.0 5.0
Children   N4 : S-R-R
OPEN   N3 S-D-D 4 2.24 6.24  N4 S-R-R 4 2.0 6.0
CLOSED   N0 S 1 0 4.0 4.0  N2 S-D 2 2 2.06 4.06  N1 S-R 3 2 3.0 5.0
-------------------------------------------
-------------------------------------------
N4 S-R-R 4 4 2.0 6.0
Children   N5 : S-R-R-D
OPEN   N5 S-R-R-D 6 1.12 7.12  N3 S-D-D 4 2.24 6.24
CLOSED   N0 S 1 0 4.0 4.0  N2 S-D 2 2 2.06 4.06  N1 S-R 3 2 3.0 5.0  N4 S-R-R 4 4 2.0 6.0
-------------------------------------------
```

Input3:

```
4
XXRR
SGRR
RRXR
RRRX
```

DLS output is

```
S-R-G 2
0.11048173904418945 the run time
```

A* output is

```
S-R-G 2
0.025463581085205078 the run time
```

For input3 map, this map is start position and goal position is next to each other and DLS and A*, even though goal node is just next, both of the algorithms visited unnecessary nodes as it is shown in below part of diagnostic screenshots.

In order to make A* search admissible and monotonic, value f(n)= g(n) + h(n) for S-RD is cheaper than actual optimal value f*(n) of S-R-G as h(n) is always underestimated. This decreases the performance of the algorithm to a certain extent.

Input4:

```
6
SRRRRR
RRRXXR
RXRRRR
RRXRXR
XRRRRR
GRRRXR
```

DLS output is

```
---------------------------------
S-RD-R-D-R-D-D-LD-L-L-G 16
0.24173736572265625 the run time
```

A* output is

```
---------------------------------
S-D-D-D-R-D-D-L-G 14
0.2768378257751465 the run time
```

For input 4, this simply shows that DLS does not always guarantee the correctness of its output while the performance of the execution could end faster than A* search. In this map, correct output is the one in A*.

Input5:

```
4
XRGR
SXRR
RRXR
RRRX
```

DLS output is

```
---------------------------------
NO-PATH
0.08961296081542969 the run time
```

A* output is

```
---------------------------------
NO-PATH
0.12031292915344238 the run time
```

Input5 is a unique map where there is no path from start node to goal node. In this situation, DLS terminates faster than A* as DLS must not

visit same node more than once. On the other hand, A* search still can check and compare the nodes visited already.

Input6:

```
6
RSRXGR
RXRXRR
RRXRXR
RRRRXR
RXRXRR
RRRRRR
```

DLS output is

```
----------------------------------------
S-L-D-D-RD-R-D-D-R-R-RU-U-U-U-LU-G 25
0.30394721031188965 the run time
```

A* output is

```
----------------------------------------
S-L-D-D-RD-R-D-D-R-R-RU-U-U-U-LU-G 25
0.3990054130554199 the run time
```

This map is slightly larger than other maps. Both DLS and A* still can return the correct path. There are not so remarkable point in this map, however, the reason that DLS can have optimal path depends on the order of direction to check, thus if the order was changed, it would not always give the optimal path, which is the main problem of DLS.

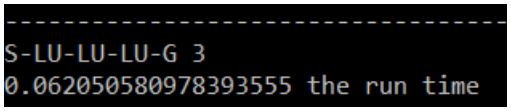
Input7:

```
4
GRRR
RRRR
RRRR
RRRS
```

DLS output is

```
S-L-L-LU-RU-LU-G 7
0.2578303813934326 the run time
```

A* output is




In this map, there are some interesting points that should be stated in this analysis.

DLS has the rule that it cannot visit same node more than once otherwise it could break its time complexity. However, it also has characteristics that it always find the path which exists within the given depth. In this map, it is possible to reach the path when the depth bound is 3 and always choose Light UP, which is dependent on the order of putting the node into OPEN list. Some said that it should regenerate the visited node to make it stable to reach the goal node when the goal is reachable within the depth bound. However, DLS must not regenerate the same node, thus there is a conflict between one constraint and another constraint.

And that is why, the runtime of DLS is much slower than that of A*.

## 6) Outputs in diagnostic mode (input1,3,7)

Input1

For DLS

```
3 [['S', 'R', 'R'], ['R', 'R', 'X'], ['R', 'R', 'G']]
-----------------------------------------------
N0 S 1 0 0 0
Children   N1 : S-R  N2 : S-D  N3 : S-RD
OPEN   N1 S-R 2 0 2  N2 S-D 2 0 2  N3 S-RD 1 0 1
CLOSED   N0 S 1 0 0 0
-----------------------------------------------
-----------------------------------------------
N3 S-RD 2 1 0 1
Children   N4 : S-RD-LD  N5 : S-RD-D
OPEN   N1 S-R 2 0 2  N2 S-D 2 0 2  N4 S-RD-LD 2 0 2  N5 S-RD-D 3 0 3
CLOSED   N0 S 1 0 0 0  N3 S-RD 2 1 0 1
-----------------------------------------------
-----------------------------------------------
N5 S-RD-D 3 3 0 3
Children   N6 : S-RD-D-R
OPEN   N1 S-R 2 0 2  N2 S-D 2 0 2  N4 S-RD-LD 2 0 2  N6 S-RD-D-R 5 0 5
CLOSED   N0 S 1 0 0 0  N3 S-RD 2 1 0 1  N5 S-RD-D 3 3 0 3
-----------------------------------------------
-----------------------------------------------
N6 S-RD-D-R-G 4 5 0 5
Children
OPEN   N1 S-R 2 0 2  N2 S-D 2 0 2  N4 S-RD-LD 2 0 2
CLOSED   N0 S 0 0 0  N3 S-RD 1 0 1  N5 S-RD-D 3 0 3  N6 S-RD-D-R-G 5 0 5
```

For A*

```
N0 S 1 0 1.41 1.41
Children   N1 : S-R  N2 : S-D  N3 : S-RD
OPEN   N1 S-R 2 1.12 3.12  N2 S-D 2 1.12 3.12  N3 S-RD 1 0.71 1.71
CLOSED   N0 S 1 0 1.41 1.41
-----------------------------------------------
-----------------------------------------------
N3 S-RD 2 1 0.71 1.71
Children   N4 : S-RD-LD  N5 : S-RD-D
OPEN   N5 S-RD-D 3 1.0 4.0  N4 S-RD-LD 2 2.0 4.0  N1 S-R 2 1.12 3.12  N2 S-D 2 1.12 3.12
CLOSED   N0 S 1 0 1.41 1.41  N3 S-RD 2 1 0.71 1.71
-----------------------------------------------
-----------------------------------------------
N2 S-D 3 2 1.12 3.12
Children
OPEN   N5 S-RD-D 3 1.0 4.0  N4 S-RD-LD 2 2.0 4.0  N1 S-R 2 1.12 3.12
CLOSED   N0 S 1 0 1.41 1.41  N3 S-RD 2 1 0.71 1.71  N2 S-D 3 2 1.12 3.12
-----------------------------------------------
-----------------------------------------------
N1 S-R 4 2 1.12 3.12
Children   N6 : S-R-R
OPEN   N6 S-R-R 4 2.0 6.0  N5 S-RD-D 3 1.0 4.0  N4 S-RD-LD 2 2.0 4.0
CLOSED   N0 S 1 0 1.41 1.41  N3 S-RD 2 1 0.71 1.71  N2 S-D 3 2 1.12 3.12  N1 S-R 4 2 1.12 3.12
-----------------------------------------------
-----------------------------------------------
N4 S-RD-LD 5 2 2.0 4.0
Children
OPEN   N6 S-R-R 4 2.0 6.0  N5 S-RD-D 3 1.0 4.0
CLOSED   N0 S 1 0 1.41 1.41  N3 S-RD 2 1 0.71 1.71  N2 S-D 3 2 1.12 3.12  N1 S-R 4 2 1.12 3.12  N4 S-RD-LD 5 2 2.0 4.0
-----------------------------------------------
-----------------------------------------------
```

Input 3

For DLS

```
-----------------------------------------------
N0 S 1 0 0 0
Children   N1 : S-R  N2 : S-D  N3 : S-RD
OPEN   N1 S-R 2 0 2  N2 S-D 2 0 2  N3 S-RD 1 0 1
CLOSED   N0 S 1 0 0 0
-----------------------------------------------
-----------------------------------------------
N3 S-RD 2 1 0 1
Children   N4 : S-RD-LD  N5 : S-RD-D
OPEN   N1 S-R 2 0 2  N2 S-D 2 0 2  N4 S-RD-LD 2 0 2  N5 S-RD-D 3 0 3
CLOSED   N0 S 1 0 0 0  N3 S-RD 2 1 0 1
-----------------------------------------------
-----------------------------------------------
N5 S-RD-D 3 3 0 3
Children   N6 : S-RD-D-R
OPEN   N1 S-R 2 0 2  N2 S-D 2 0 2  N4 S-RD-LD 2 0 2  N6 S-RD-D-R 5 0 5
CLOSED   N0 S 1 0 0 0  N3 S-RD 2 1 0 1  N5 S-RD-D 3 3 0 3
-----------------------------------------------
-----------------------------------------------
N6 S-RD-D-R 4 5 0 5
Children
OPEN   N1 S-R 2 0 2  N2 S-D 2 0 2  N4 S-RD-LD 2 0 2
CLOSED   N0 S 1 0 0 0  N3 S-RD 2 1 0 1  N5 S-RD-D 3 3 0 3  N6 S-RD-D-R 4 5 0 5
-----------------------------------------------
-----------------------------------------------
```

For A*

```
N0 S 1 0 1.0 1.0
Children   N1 : S-R  N2 : S-D  N3 : S-RD
OPEN   N2 S-D 2 0.71 2.71  N1 S-R 2 0.0 2.0  N3 S-RD 1 1.0 2.0
CLOSED   N0 S 1 0 1.0 1.0
-----------------------------------------------
-----------------------------------------------
N3 S-RD 2 1 1.0 2.0
Children   N4 : S-RD-LD  N5 : S-RD-D
OPEN   N5 S-RD-D 3 2.0 5.0  N4 S-RD-LD 2 1.12 3.12  N2 S-D 2 0.71 2.71  N1 S-R 2 0.0 2.0
CLOSED   N0 S 1 0 1.0 1.0  N3 S-RD 2 1 1.0 2.0
-----------------------------------------------
-----------------------------------------------
N1 S-R-G 3 2 0.0 2.0
Children   N4 : S-RD-LD  N5 : S-RD-D
OPEN   N5 S-RD-D 3 2.0 5.0  N4 S-RD-LD 2 1.12 3.12  N2 S-D 2 0.71 2.71
CLOSED   N0 S 1 0 1.0 1.0  N3 S-RD 2 1 1.0 2.0  N1 S-R-G 3 2 0.0 2.0
-----------------------------------------------
```

Input 7:

For DLS

```
------------------------------------------------
N0 S 1 0 0 0
Children   N1 : S-LU  N2 : S-U  N3 : S-L
OPEN   N1 S-LU 1 0 1  N2 S-U 2 0 2  N3 S-L 2 0 2
CLOSED   N0 S 1 0 0 0
------------------------------------------------
------------------------------------------------
N3 S-L 2 2 0 2
Children   N4 : S-L-LU  N5 : S-L-L
OPEN   N1 S-LU 1 0 1  N2 S-U 2 0 2  N4 S-L-LU 3 0 3  N5 S-L-L 4 0 4
CLOSED   N0 S 1 0 0 0  N3 S-L 2 2 0 2
------------------------------------------------
------------------------------------------------
N5 S-L-L 3 4 0 4
Children   N6 : S-L-L-LU  N7 : S-L-L-L
OPEN   N1 S-LU 1 0 1  N2 S-U 2 0 2  N4 S-L-LU 3 0 3  N6 S-L-L-LU 5 0 5  N7 S-L-L-L 6 0 6
CLOSED   N0 S 1 0 0 0  N3 S-L 2 2 0 2  N5 S-L-L 3 4 0 4
------------------------------------------------
------------------------------------------------
N7 S-L-L-L 4 6 0 6
Children
OPEN   N1 S-LU 1 0 1  N2 S-U 2 0 2  N4 S-L-LU 3 0 3  N6 S-L-L-LU 5 0 5
CLOSED   N0 S 1 0 0 0  N3 S-L 2 2 0 2  N5 S-L-L 3 4 0 4  N7 S-L-L-L 4 6 0 6
------------------------------------------------
------------------------------------------------
N6 S-L-L-LU 5 5 0 5
Children   N8 : S-L-L-LU-U  N9 : S-L-L-LU-RU
OPEN   N1 S-LU 1 0 1  N2 S-U 2 0 2  N4 S-L-LU 3 0 3  N8 S-L-L-LU-U 7 0 7  N9 S-L-L-LU-RU 6 0 6
CLOSED   N0 S 1 0 0 0  N3 S-L 2 2 0 2  N5 S-L-L 3 4 0 4  N7 S-L-L-L 4 6 0 6  N6 S-L-L-LU 5 5 0 5
------------------------------------------------
------------------------------------------------
N9 S-L-L-LU-RU 6 6 0 6
Children   N10 : S-L-L-LU-RU-LU  N11 : S-L-L-LU-RU-U  N12 : S-L-L-LU-RU-RU  N13 : S-L-L-LU-RU-R
OPEN   N1 S-LU 1 0 1  N2 S-U 2 0 2  N4 S-L-LU 3 0 3  N8 S-L-L-LU-U 7 0 7  N10 S-L-L-LU-RU-LU 7 0 7  N11 S-L-L-LU-RU-U 8 0 8  N12 S-L-L-LU-RU-RU 7 0 7  N13 S-L-L-LU-RU-R 8 0 8
CLOSED   N0 S 1 0 0 0  N3 S-L 2 2 0 2  N5 S-L-L 3 4 0 4  N7 S-L-L-L 4 6 0 6  N6 S-L-L-LU 5 5 0 5  N9 S-L-L-LU-RU 6 6 0 6
------------------------------------------------
------------------------------------------------
```

For A*

```
------------------------------------------------
N0 S 1 0 2.12 2.12
Children   N1 : S-LU  N2 : S-U  N3 : S-L
OPEN   N2 S-U 2 1.8 3.8  N3 S-L 2 1.8 3.8  N1 S-LU 1 1.41 2.41
CLOSED   N0 S 1 0 2.12 2.12
------------------------------------------------
------------------------------------------------
N1 S-LU 2 1 1.41 2.41
Children   N4 : S-LU-LU  N5 : S-LU-U  N6 : S-LU-RU  N7 : S-LU-L  N8 : S-LU-LD
OPEN   N5 S-LU-U 3 1.12 4.12  N7 S-LU-L 3 1.12 4.12  N2 S-U 2 1.8 3.8  N3 S-L 2 1.8 3.8  N6 S-LU-RU 2 1.58 3.58  N8 S-LU-LD 2 1.58 3.58  N4 S-LU-LU 2 0.71 2.71
CLOSED   N0 S 1 0 2.12 2.12  N1 S-LU 2 1 1.41 2.41
------------------------------------------------
------------------------------------------------
N4 S-LU-LU 3 2 0.71 2.71
Children   N9 : S-LU-LU-LU  N10 : S-LU-LU-U  N11 : S-LU-LU-RU  N12 : S-LU-LU-L  N13 : S-LU-LU-LD
OPEN   N10 S-LU-LU-U 4 1.0 5.0  N12 S-LU-LU-L 4 1.0 5.0  N11 S-LU-LU-RU 3 2.0 5.0  N13 S-LU-LU-LD 3 2.0 5.0  N5 S-LU-U 3 1.12 4.12  N7 S-LU-L 3 1.12 4.12  N2 S-U 2 1.8 3.8  N3 S-L 2 1.8 3.8  N6 S-LU-RU 2 1.58 3.
58  N8 S-LU-LD 2 1.58 3.58  N9 S-LU-LU-LU 3 0.0 3.0
CLOSED   N0 S 1 0 2.12 2.12  N1 S-LU 2 1 1.41 2.41  N4 S-LU-LU 3 2 0.71 2.71
------------------------------------------------
------------------------------------------------
N9 S-LU-LU-LU-G 4 3 0.0 3.0
Children   N9 : S-LU-LU-LU  N10 : S-LU-LU-U  N11 : S-LU-LU-RU  N12 : S-LU-LU-L  N13 : S-LU-LU-LD
OPEN   N10 S-LU-LU-U 4 1.0 5.0  N12 S-LU-LU-L 4 1.0 5.0  N11 S-LU-LU-RU 3 2.0 5.0  N13 S-LU-LU-LD 3 2.0 5.0  N5 S-LU-U 3 1.12 4.12  N7 S-LU-L 3 1.12 4.12  N2 S-U 2 1.8 3.8  N3 S-L 2 1.8 3.8  N6 S-LU-RU 2 1.58 3.
58  N8 S-LU-LD 2 1.58 3.58
CLOSED   N0 S 1 0 2.12 2.12  N1 S-LU 2 1 1.41 2.41  N4 S-LU-LU 3 2 0.71 2.71  N9 S-LU-LU-LU-G 4 3 0.0 3.0
------------------------------------------------
```