CS 541: Artificial Intelligence
Programming Assignment #1

In this assignment, I have implemented both Best First Search and A* search algorithms for N puzzle problems, where n= 8,15. Three heuristics have been used, they are Manhattan Distance, Euclidean Distance and Misplaced Tiles.

Note: The solution paths are printed in reverse order.

8-puzzle Problem:

Best-First-Search:

Number of trails = 5

Heuristic1: Manhattan Distance	*Heuristic2: Euclidean Distance*	*Heuristic3: Misplaced Tiles*
Initial State1: *[8,1,3,4,'b',2,7,6,5]* *Final solution path:* *[1,2,3,4,5,6,7,8,b]* *[1, 2, 3, 4, 5, 'b', 7, 8, 6]* *[1, 2, 3, 4, 'b', 5, 7, 8, 6]* *[1, 2, 3, 4, 8, 5, 7, 'b', 6]* *[1, 2, 3, 4, 8, 5, 'b', 7, 6]* *[1, 2, 3, 'b', 8, 5, 4, 7, 6]* *[1, 2, 3, 8, 'b', 5, 4, 7, 6]*	*Initial State1:* *[8,1,3,4,'b',2,7,6,5]* *Final solution path:* *[1,2,3,4,5,6,7,8,b]* *[1, 2, 3, 4, 5, 6, 7, 'b', 8]* *[1, 2, 3, 4, 'b', 6, 7, 5, 8]* *[1, 'b', 3, 4, 2, 6, 7, 5, 8]* *[1, 3, 'b', 4, 2, 6, 7, 5, 8]* *[1, 3, 6, 4, 2, 'b', 7, 5, 8]* *[1, 3, 6, 4, 'b', 2, 7, 5, 8]*	*Initial State1:* *[8,1,3,4,'b',2,7,6,5]* *Final solution path:* *[1,2,3,4,5,6,7,8,b]* *[1, 2, 3, 4, 5, 6, 7, 'b', 8]* *[1, 2, 3, 4, 'b', 6, 7, 5, 8]* *[1, 'b', 3, 4, 2, 6, 7, 5, 8]* *[1, 3, 'b', 4, 2, 6, 7, 5, 8]* *[1, 3, 6, 4, 2, 'b', 7, 5, 8]* *[1, 3, 6, 4, 'b', 2, 7, 5, 8]*

[1, 'b', 3, 8, 2, 5, 4, 7, 6] *['b', 1, 3, 8, 2, 5, 4, 7, 6]* *[8, 1, 3, 'b', 2, 5, 4, 7, 6]* *[8, 1, 3, 4, 2, 5, 'b', 7, 6]* *[8, 1, 3, 4, 2, 5, 7, 'b', 6]* *[8, 1, 3, 4, 2, 5, 7, 6, 'b']* *[8, 1, 3, 4, 2, 'b', 7, 6, 5]* *[8, 1, 3, 4, 'b', 2, 7, 6, 5]* *Steps: 15*	*[1, 3, 6, 4, 5, 2, 7, 'b', 8]* *[1, 3, 6, 4, 5, 2, 7, 8, 'b']* *[1, 3, 6, 4, 5, 'b', 7, 8, 2]* *[1, 3, 'b', 4, 5, 6, 7, 8, 2]* *[1, 'b', 3, 4, 5, 6, 7, 8, 2]* *[1, 5, 3, 4, 'b', 6, 7, 8, 2]* *[1, 5, 3, 4, 8, 6, 7, 'b', 2]* *[1, 5, 3, 4, 8, 6, 'b', 7, 2]* *[1, 5, 3, 'b', 8, 6, 4, 7, 2]* *[1, 5, 3, 8, 'b', 6, 4, 7, 2]* *[1, 'b', 3, 8, 5, 6, 4, 7, 2]* *['b', 1, 3, 8, 5, 6, 4, 7, 2]* *[8, 1, 3, 'b', 5, 6, 4, 7, 2]* *Steps: 29*	*[1, 3, 6, 4, 5, 2, 7, 'b', 8]* *[1, 3, 6, 4, 5, 2, 7, 8, 'b']* *[1, 3, 6, 4, 5, 'b', 7, 8, 2]* *[1, 3, 'b', 4, 5, 6, 7, 8, 2]* *[1, 'b', 3, 4, 5, 6, 7, 8, 2]* *[1, 5, 3, 4, 'b', 6, 7, 8, 2]* *[1, 5, 3, 4, 8, 6, 7, 'b', 2]* *[1, 5, 3, 4, 8, 6, 'b', 7, 2]* *[1, 5, 3, 'b', 8, 6, 4, 7, 2]* *[1, 5, 3, 8, 'b', 6, 4, 7, 2]* *[1, 'b', 3, 8, 5, 6, 4, 7, 2]* *['b', 1, 3, 8, 5, 6, 4, 7, 2]* *[8, 1, 3, 'b', 5, 6, 4, 7, 2]* *Steps: 27*

Initial State 2: [7, 1, 3, 4, 'b', 2, 5, 6, 8]	*Initial State 2: [7, 1, 3, 4, 'b', 2, 5, 6, 8]*	*Initial State 2: [7, 1, 3, 4, 'b', 2, 5, 6, 8]*
Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']	*Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']*	*Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']*
[1, 2, 3, 4, 5, 6, 7, 'b', 8]	*[1, 2, 3, 4, 5, 'b', 7, 8, 6]*	*[1, 2, 3, 4, 5, 6, 7, 'b', 8]*
[1, 2, 3, 4, 5, 6, 'b', 7, 8]	*[1, 2, 3, 4, 'b', 5, 7, 8, 6]*	*[1, 2, 3, 4, 'b', 6, 7, 5, 8]*
[1, 2, 3, 'b', 5, 6, 4, 7, 8]	*[1, 2, 3, 'b', 4, 5, 7, 8, 6]*	*[1, 'b', 3, 4, 2, 6, 7, 5, 8]*
[1, 2, 3, 5, 'b', 6, 4, 7, 8]	*['b', 2, 3, 1, 4, 5, 7, 8, 6]*	*[1, 3, 'b', 4, 2, 6, 7, 5, 8]*
[1, 'b', 3, 5, 2, 6, 4, 7, 8]	*[2, 'b', 3, 1, 4, 5, 7, 8, 6]*	*[1, 3, 6, 4, 2, 'b', 7, 5, 8]*

['b', 1, 3, 5, 2, 6, 4, 7, 8] *[5, 1, 3, 'b', 2, 6, 4, 7, 8]* *[5, 1, 3, 4, 2, 6, 'b', 7, 8]* *[5, 1, 3, 4, 2, 6, 7, 'b', 8]* *[5, 1, 3, 4, 2, 6, 7, 8, 'b']* *[5, 1, 3, 4, 2, 'b', 7, 8, 6]* *[5, 1, 3, 4, 'b', 2, 7, 8, 6]* *[1, 3, 2, 5, 4, 6, 7, 'b', 8]* *[1, 3, 2, 5, 'b', 6, 7, 4, 8]* *[1, 3, 2, 'b', 5, 6, 7, 4, 8]* *[1, 3, 2, 7, 5, 6, 'b', 4, 8]* *[7, 1, 3, 'b', 6, 2, 4, 5, 8]* *[7, 1, 3, 4, 6, 2, 'b', 5, 8]* *[7, 1, 3, 4, 6, 2, 5, 'b', 8]* *[7, 1, 3, 4, 'b', 2, 5, 6, 8]* *steps: 33*	*[2, 4, 3, 1, 'b', 5, 7, 8, 6]* *[2, 4, 3, 1, 5, 'b', 7, 8, 6]* *[2, 4, 'b', 1, 5, 3, 7, 8, 6]* *[2, 'b', 4, 1, 5, 3, 7, 8, 6]* *['b', 2, 4, 1, 5, 3, 7, 8, 6]* *[1, 2, 4, 'b', 5, 3, 7, 8, 6]* *[1, 2, 4, 5, 'b', 3, 7, 8, 6]* *[1, 'b', 4, 5, 2, 3, 7, 8, 6]* *[1, 4, 'b', 5, 2, 3, 7, 8, 6]* *[1, 4, 3, 5, 2, 'b', 7, 8, 6]* *[1, 4, 3, 5, 'b', 2, 7, 8, 6]* *[1, 'b', 3, 5, 4, 2, 7, 8, 6]* *[1, 3, 'b', 5, 4, 2, 7, 8, 6]* *[1, 3, 2, 5, 4, 'b', 7, 8, 6]* *[1, 3, 2, 5, 4, 6, 7, 8, 'b']* *[1, 3, 2, 5, 4, 6, 7, 'b', 8]* *[1, 3, 2, 5, 'b', 6, 7, 4, 8]* *[1, 'b', 2, 5, 3, 6, 7, 4, 8]* *[1, 2, 'b', 5, 3, 6, 7, 4, 8]* *[1, 4, 3, 7, 6, 2, 5, 8, 'b']* *[1, 4, 3, 7, 6, 'b', 5, 8, 2]* *[1, 4, 3, 7, 'b', 6, 5, 8, 2]* *[1, 'b', 3, 7, 4, 6, 5, 8, 2]* *['b', 1, 3, 7, 4, 6, 5, 8, 2]* *[7, 1, 3, 'b', 4, 6, 5, 8, 2]*	*[1, 3, 6, 4, 'b', 2, 7, 5, 8]* *[1, 3, 6, 4, 5, 2, 7, 'b', 8]* *[1, 3, 6, 4, 5, 2, 7, 8, 'b']* *[1, 3, 6, 4, 5, 'b', 7, 8, 2]* *[1, 3, 'b', 4, 5, 6, 7, 8, 2]* *[1, 'b', 3, 4, 5, 6, 7, 8, 2]* *[1, 5, 3, 4, 'b', 6, 7, 8, 2]* *[1, 5, 3, 'b', 4, 6, 7, 8, 2]* *[1, 5, 3, 7, 4, 6, 'b', 8, 2]* *[1, 5, 3, 7, 4, 6, 8, 'b', 2]* *[1, 5, 3, 7, 'b', 6, 8, 4, 2]* *[1, 'b', 3, 7, 5, 6, 8, 4, 2]* *['b', 1, 3, 7, 5, 6, 8, 4, 2]* *[7, 1, 3, 'b', 4, 6, 5, 8, 2]* *[7, 1, 3, 4, 'b', 6, 5, 8, 2]* *[7, 1, 3, 4, 6, 'b', 5, 8, 2]* *[7, 1, 3, 4, 6, 2, 5, 8, 'b']* *[7, 1, 3, 4, 6, 2, 5, 'b', 8]* *[7, 1, 3, 4, 'b', 2, 5, 6, 8]* *steps: 29*

	[7, 1, 3, 4, 'b', 6, 5, 8, 2] *[7, 1, 3, 4, 6, 'b', 5, 8, 2]* *[7, 1, 3, 4, 6, 2, 5, 8, 'b']* *[7, 1, 3, 4, 6, 2, 5, 'b', 8]* *[7, 1, 3, 4, 'b', 2, 5, 6, 8]* *steps: 65*	

Initial State 3: [1, 6, 3, 4, 'b', 2, 5, 7, 8] *Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']* *[1, 2, 3, 4, 5, 6, 7, 'b', 8]* *[1, 2, 3, 4, 5, 6, 'b', 7, 8]* *[1, 2, 3, 'b', 5, 6, 4, 7, 8]* *[1, 2, 3, 5, 'b', 6, 4, 7, 8]* *[1, 'b', 3, 5, 2, 6, 4, 7, 8]* *['b', 1, 3, 5, 2, 6, 4, 7, 8]* *[5, 1, 3, 'b', 2, 6, 4, 7, 8]* *[5, 1, 3, 4, 2, 6, 'b', 7, 8]* *[5, 1, 3, 4, 2, 6, 7, 'b', 8]* *[5, 1, 3, 4, 2, 6, 7, 8, 'b']*	*Initial State 3: [1, 6, 3, 4, 'b', 2, 5, 7, 8]* *Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']* *[1, 2, 3, 4, 5, 'b', 7, 8, 6]* *[1, 2, 3, 4, 'b', 5, 7, 8, 6]* *[1, 2, 3, 'b', 4, 5, 7, 8, 6]* *['b', 2, 3, 1, 4, 5, 7, 8, 6]* *[2, 'b', 3, 1, 4, 5, 7, 8, 6]* *[2, 4, 3, 1, 'b', 5, 7, 8, 6]* *[2, 4, 3, 1, 5, 'b', 7, 8, 6]* *[2, 4, 'b', 1, 5, 3, 7, 8, 6]* *[2, 'b', 4, 1, 5, 3, 7, 8, 6]* *[1, 3, 2, 5, 4, 6, 7, 8, 'b']*	*Initial State 3: [1, 6, 3, 4, 'b', 2, 5, 7, 8]* *Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']* *[1, 2, 3, 4, 5, 6, 7, 'b', 8]* *[1, 2, 3, 4, 'b', 6, 7, 5, 8]* *[1, 'b', 3, 4, 2, 6, 7, 5, 8]* *[1, 3, 'b', 4, 2, 6, 7, 5, 8]* *[1, 3, 6, 4, 2, 'b', 7, 5, 8]* *[1, 3, 6, 4, 'b', 2, 7, 5, 8]* *[1, 3, 6, 4, 5, 2, 7, 'b', 8]* *[1, 3, 6, 4, 5, 2, 7, 8, 'b']* *[1, 3, 6, 4, 5, 'b', 7, 8, 2]*

[5, 1, 3, 4, 2, 'b', 7, 8, 6] *[5, 1, 3, 4, 'b', 2, 7, 8, 6]* *[5, 1, 3, 'b', 4, 2, 7, 8, 6]* *['b', 1, 3, 5, 4, 2, 7, 8, 6]* *[1, 'b', 3, 5, 4, 2, 7, 8, 6]* *[1, 3, 2, 5, 'b', 6, 7, 4, 8]* *[1, 'b', 2, 5, 3, 6, 7, 4, 8]* *[1, 2, 'b', 5, 3, 6, 7, 4, 8]* *[1, 2, 6, 5, 3, 'b', 7, 4, 8]* *[1, 2, 6, 5, 'b', 3, 7, 4, 8]* *[1, 2, 6, 5, 4, 3, 7, 'b', 8]* *[1, 2, 6, 5, 4, 3, 'b', 7, 8]* *[1, 2, 6, 'b', 4, 3, 5, 7, 8]* *[1, 2, 6, 4, 'b', 3, 5, 7, 8]* *[1, 'b', 6, 4, 2, 3, 5, 7, 8]* *[1, 6, 'b', 4, 2, 3, 5, 7, 8]* *[1, 6, 3, 4, 2, 'b', 5, 7, 8]* *[1, 6, 3, 4, 'b', 2, 5, 7, 8]* *steps: 33*	*[1, 3, 2, 5, 4, 6, 7, 'b', 8]* *[1, 3, 2, 5, 'b', 6, 7, 4, 8]* *[1, 'b', 2, 5, 3, 6, 7, 4, 8]* *[1, 2, 'b', 5, 3, 6, 7, 4, 8]* *[1, 2, 6, 5, 3, 'b', 7, 4, 8]* *[1, 2, 6, 5, 'b', 3, 7, 4, 8]* *[1, 2, 6, 5, 4, 3, 7, 'b', 8]* *[1, 2, 6, 5, 4, 3, 'b', 7, 8]* *[1, 2, 6, 'b', 4, 3, 5, 7, 8]* *[1, 2, 6, 4, 'b', 3, 5, 7, 8]* *[1, 'b', 6, 4, 2, 3, 5, 7, 8]* *[1, 6, 'b', 4, 2, 3, 5, 7, 8]* *[1, 6, 3, 4, 2, 'b', 5, 7, 8]* *[1, 6, 3, 4, 'b', 2, 5, 7, 8]* *steps: 35*	*[1, 3, 'b', 4, 5, 6, 7, 8, 2]* *[1, 'b', 3, 4, 5, 6, 7, 8, 2]* *[1, 5, 3, 4, 'b', 6, 7, 8, 2]* *[1, 5, 3, 4, 6, 'b', 7, 8, 2]* *[1, 5, 'b', 4, 6, 3, 7, 8, 2]* *[1, 'b', 5, 4, 6, 3, 7, 8, 2]* *[1, 6, 5, 4, 'b', 3, 7, 8, 2]* *[1, 6, 5, 'b', 4, 3, 7, 8, 2]* *['b', 6, 5, 1, 4, 3, 7, 8, 2]* *['b', 6, 3, 1, 5, 4, 7, 8, 2]* *[1, 6, 3, 'b', 5, 4, 7, 8, 2]* *[1, 6, 3, 5, 'b', 4, 7, 8, 2]* *[1, 6, 3, 5, 4, 'b', 7, 8, 2]* *[1, 6, 3, 5, 4, 2, 7, 8, 'b']* *[1, 6, 3, 5, 4, 2, 7, 'b', 8]* *[1, 6, 3, 5, 4, 2, 'b', 7, 8]* *[1, 6, 3, 'b', 4, 2, 5, 7, 8]* *[1, 6, 3, 4, 'b', 2, 5, 7, 8]* *steps: 33*

Initial State 4: [1, 2, 3, 4, 5, 6, 'b', 7, 8] *Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']* *[1, 2, 3, 4, 5, 6, 7, 'b', 8]* *[1, 2, 3, 4, 5, 6, 'b', 7, 8]* *steps: 3*	*Initial State 4: [1, 2, 3, 4, 5, 6, 'b', 7, 8]* *Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']* *[1, 2, 3, 4, 5, 6, 7, 'b', 8]* *[1, 2, 3, 4, 5, 6, 'b', 7, 8]* *steps: 3*	*Initial State 4: [1, 2, 3, 4, 5, 6, 'b', 7, 8]* *Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']* *[1, 2, 3, 4, 5, 6, 7, 'b', 8]* *[1, 2, 3, 4, 5, 6, 'b', 7, 8]* *steps: 3*
Initial State 5: [1, 2, 3, 4, 5, 6, 7, 'b', 8] *Solution Path: [1, 2, 3, 4, 5, 6, 7, 'b', 8]* *[1, 2, 3, 4, 5, 6, 7, 8, 'b']* *steps: 2*	*Initial State 5: [1, 2, 3, 4, 5, 6, 7, 'b', 8]* *Solution Path: [1, 2, 3, 4, 5, 6, 7, 'b', 8]* *[1, 2, 3, 4, 5, 6, 7, 8, 'b']* *steps: 2*	*Initial State5: [1, 2, 3, 4, 5, 6, 7, 'b', 8]* *Solution Path: [1, 2, 3, 4, 5, 6, 7, 'b', 8]* *[1, 2, 3, 4, 5, 6, 7, 8, 'b']* *steps: 2*
Average number of steps: : (2+15+33+33+3)/5 = 17.2	*Average number of steps: : (2+29+65+35+3)/5 = 26.8*	*Average number of steps: : (2+27+29+33+3)/5 = 18.8*

A* search :

Number of trails = 5

Heuristic1: Manhattan Distance	*Heuristic2: Euclidean Distance*	*Heuristic3: Misplaced Tiles*

Initial State 1: *[8,1,3,4,'b',2,7,6,5]* *Final solution path:* *[1,2,3,4,5,6,7,8,b]*	*Initial State1 :* *[8,1,3,4,'b',2,7,6,5]* *Final solution path:* *[1,2,3,4,5,6,7,8,b]*	*Initial State1:* *[8,1,3,4,'b',2,7,6,5]* *Final solution path:* *[1,2,3,4,5,6,7,8,b]*

[1, 2, 3, 4, 5, 'b', 7, 8, 6] *[1, 2, 3, 4, 'b', 5, 7, 8, 6]* *[1, 2, 3, 4, 8, 5, 7, 'b', 6]* *[1, 2, 3, 4, 8, 5, 'b', 7, 6]* *[1, 2, 3, 'b', 8, 5, 4, 7, 6]* *[1, 2, 3, 8, 'b', 5, 4, 7, 6]* *[1, 'b', 3, 8, 2, 5, 4, 7, 6]* *['b', 1, 3, 8, 2, 5, 4, 7, 6]* *[8, 1, 3, 'b', 2, 5, 4, 7, 6]* *[8, 1, 3, 4, 2, 5, 'b', 7, 6]* *[8, 1, 3, 4, 2, 5, 7, 'b', 6]* *[8, 1, 3, 4, 2, 5, 7, 6, 'b']* *[8, 1, 3, 4, 2, 'b', 7, 6, 5]* *[8, 1, 3, 4, 'b', 2, 7, 6, 5]* *Steps: 15*	*[1, 2, 3, 4, 5, 6, 7, 8, 'b']* *[1, 2, 3, 4, 5, 'b', 7, 8, 6]* *[1, 2, 3, 4, 'b', 5, 7, 8, 6]* *[1, 2, 3, 4, 8, 5, 7, 'b', 6]* *[1, 2, 3, 4, 8, 5, 'b', 7, 6]* *[1, 2, 3, 'b', 8, 5, 4, 7, 6]* *[1, 2, 3, 8, 'b', 5, 4, 7, 6]* *[1, 'b', 3, 8, 2, 5, 4, 7, 6]* *['b', 1, 3, 8, 2, 5, 4, 7, 6]* *[8, 1, 3, 'b', 2, 5, 4, 7, 6]* *[8, 1, 3, 4, 2, 5, 'b', 7, 6]* *[8, 1, 3, 4, 2, 5, 7, 'b', 6]* *[8, 1, 3, 4, 2, 5, 7, 6, 'b']* *[8, 1, 3, 4, 2, 'b', 7, 6, 5]* *[8, 1, 3, 4, 'b', 2, 7, 6, 5]* *Steps: 15*	*[1, 2, 3, 4, 5, 6, 7, 8, 'b']* *[1, 2, 3, 4, 5, 'b', 7, 8, 6]* *[1, 2, 3, 4, 'b', 5, 7, 8, 6]* *[1, 2, 3, 4, 8, 5, 7, 'b', 6]* *[1, 2, 3, 4, 8, 5, 'b', 7, 6]* *[1, 2, 3, 'b', 8, 5, 4, 7, 6]* *[1, 2, 3, 8, 'b', 5, 4, 7, 6]* *[1, 'b', 3, 8, 2, 5, 4, 7, 6]* *['b', 1, 3, 8, 2, 5, 4, 7, 6]* *[8, 1, 3, 'b', 2, 5, 4, 7, 6]* *[8, 1, 3, 4, 2, 5, 'b', 7, 6]* *[8, 1, 3, 4, 2, 5, 7, 'b', 6]* *[8, 1, 3, 4, 2, 5, 7, 6, 'b']* *[8, 1, 3, 4, 2, 'b', 7, 6, 5]* *[8, 1, 3, 4, 'b', 2, 7, 6, 5]* *Steps: 15*

Initial State 2: [7, 1, 3, 4, 'b', 2, 5, 6, 8] *Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']* *[1, 2, 3, 4, 5, 6, 7, 'b', 8]*	*Initial State 2: [7, 1, 3, 4, 'b', 2, 5, 6, 8]* *Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']* *[1, 2, 3, 4, 5, 6, 7, 'b', 8]*	*Initial State 2: [7, 1, 3, 4, 'b', 2, 5, 6, 8]* *Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']* *[1, 2, 3, 4, 5, 6, 7, 'b', 8]*

[1, 2, 3, 4, 'b', 6, 7, 5, 8]	*[1, 2, 3, 4, 'b', 6, 7, 5, 8]*	*[1, 2, 3, 4, 'b', 6, 7, 5, 8]*
[1, 'b', 3, 4, 2, 6, 7, 5, 8]	*[1, 'b', 3, 4, 2, 6, 7, 5, 8]*	*[1, 'b', 3, 4, 2, 6, 7, 5, 8]*
[1, 3, 'b', 4, 2, 6, 7, 5, 8]	*[1, 3, 'b', 4, 2, 6, 7, 5, 8]*	*[1, 3, 'b', 4, 2, 6, 7, 5, 8]*
[1, 3, 6, 4, 2, 'b', 7, 5, 8]	*[1, 3, 6, 4, 2, 'b', 7, 5, 8]*	*[1, 3, 6, 4, 2, 'b', 7, 5, 8]*
[1, 3, 6, 4, 'b', 2, 7, 5, 8]	*[1, 3, 6, 4, 'b', 2, 7, 5, 8]*	*[1, 3, 6, 4, 'b', 2, 7, 5, 8]*
[1, 3, 6, 'b', 4, 2, 7, 5, 8]	*[1, 3, 6, 'b', 4, 2, 7, 5, 8]*	*[1, 3, 6, 'b', 4, 2, 7, 5, 8]*
[1, 3, 6, 7, 4, 2, 'b', 5, 8]	*[1, 3, 6, 7, 4, 2, 'b', 5, 8]*	*[1, 3, 6, 7, 4, 2, 'b', 5, 8]*
[1, 3, 6, 7, 4, 2, 5, 'b', 8]	*[1, 3, 6, 7, 4, 2, 5, 'b', 8]*	*[1, 3, 6, 7, 4, 2, 5, 'b', 8]*
[1, 3, 6, 7, 4, 2, 5, 8, 'b']	*[1, 3, 6, 7, 4, 2, 5, 8, 'b']*	*[1, 3, 6, 7, 4, 2, 5, 8, 'b']*
[1, 3, 6, 7, 4, 'b', 5, 8, 2]	*[1, 3, 6, 7, 4, 'b', 5, 8, 2]*	*[1, 3, 6, 7, 4, 'b', 5, 8, 2]*
[1, 3, 'b', 7, 4, 6, 5, 8, 2]	*[1, 3, 'b', 7, 4, 6, 5, 8, 2]*	*[1, 3, 'b', 7, 4, 6, 5, 8, 2]*
[1, 'b', 3, 7, 4, 6, 5, 8, 2]	*[1, 'b', 3, 7, 4, 6, 5, 8, 2]*	*[1, 'b', 3, 7, 4, 6, 5, 8, 2]*
['b', 1, 3, 7, 4, 6, 5, 8, 2]	*['b', 1, 3, 7, 4, 6, 5, 8, 2]*	*['b', 1, 3, 7, 4, 6, 5, 8, 2]*
[7, 1, 3, 'b', 4, 6, 5, 8, 2]	*[7, 1, 3, 'b', 4, 6, 5, 8, 2]*	*[7, 1, 3, 'b', 4, 6, 5, 8, 2]*
[7, 1, 3, 4, 'b', 6, 5, 8, 2]	*[7, 1, 3, 4, 'b', 6, 5, 8, 2]*	*[7, 1, 3, 4, 'b', 6, 5, 8, 2]*
[7, 1, 3, 4, 6, 'b', 5, 8, 2]	*[7, 1, 3, 4, 6, 'b', 5, 8, 2]*	*[7, 1, 3, 4, 6, 'b', 5, 8, 2]*
[7, 1, 3, 4, 6, 2, 5, 8, 'b']	*[7, 1, 3, 4, 6, 2, 5, 8, 'b']*	*[7, 1, 3, 4, 6, 2, 5, 8, 'b']*
[7, 1, 3, 4, 6, 2, 5, 'b', 8]	*[7, 1, 3, 4, 6, 2, 5, 'b', 8]*	*[7, 1, 3, 4, 6, 2, 5, 'b', 8]*
[7, 1, 3, 4, 'b', 2, 5, 6, 8] *steps: 21*	*[7, 1, 3, 4, 'b', 2, 5, 6, 8]* *steps: 21*	*[7, 1, 3, 4, 'b', 2, 5, 6, 8]* *steps: 21*

Initial State 3: [1, 6, 3, 4, 'b', 2, 5, 7, 8] *Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']*	*Initial State 3: [1, 6, 3, 4, 'b', 2, 5, 7, 8]* *Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']*	*Initial State 3: [1, 6, 3, 4, 'b', 2, 5, 7, 8]* *Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']*

[1, 2, 3, 4, 5, 'b', 7, 8, 6] *[1, 2, 'b', 4, 5, 3, 7, 8, 6]* *[1, 'b', 2, 4, 5, 3, 7, 8, 6]* *['b', 1, 2, 4, 5, 3, 7, 8, 6]* *[4, 1, 2, 'b', 5, 3, 7, 8, 6]* *[4, 1, 2, 5, 'b', 3, 7, 8, 6]* *[4, 1, 2, 5, 3, 'b', 7, 8, 6]* *[4, 1, 2, 5, 3, 6, 7, 8, 'b']* *[4, 1, 2, 5, 3, 6, 7, 'b', 8]* *[4, 1, 2, 5, 3, 6, 'b', 7, 8]* *[4, 1, 2, 'b', 3, 6, 5, 7, 8]* *['b', 1, 2, 4, 3, 6, 5, 7, 8]* *[1, 'b', 2, 4, 3, 6, 5, 7, 8]* *[1, 3, 2, 4, 'b', 6, 5, 7, 8]* *[1, 3, 2, 4, 6, 'b', 5, 7, 8]* *[1, 3, 'b', 4, 6, 2, 5, 7, 8]* *[1, 'b', 3, 4, 6, 2, 5, 7, 8]* *[1, 6, 3, 4, 'b', 2, 5, 7, 8]* *steps: 19*	*[1, 2, 3, 4, 5, 'b', 7, 8, 6]* *[1, 2, 'b', 4, 5, 3, 7, 8, 6]* *[1, 'b', 2, 4, 5, 3, 7, 8, 6]* *['b', 1, 2, 4, 5, 3, 7, 8, 6]* *[4, 1, 2, 'b', 5, 3, 7, 8, 6]* *[4, 1, 2, 5, 'b', 3, 7, 8, 6]* *[4, 1, 2, 5, 3, 'b', 7, 8, 6]* *[4, 1, 2, 5, 3, 6, 7, 8, 'b']* *[4, 1, 2, 5, 3, 6, 7, 'b', 8]* *[4, 1, 2, 5, 3, 6, 'b', 7, 8]* *[4, 1, 2, 'b', 3, 6, 5, 7, 8]* *['b', 1, 2, 4, 3, 6, 5, 7, 8]* *[1, 'b', 2, 4, 3, 6, 5, 7, 8]* *[1, 3, 2, 4, 'b', 6, 5, 7, 8]* *[1, 3, 2, 4, 6, 'b', 5, 7, 8]* *[1, 3, 'b', 4, 6, 2, 5, 7, 8]* *[1, 'b', 3, 4, 6, 2, 5, 7, 8]* *[1, 6, 3, 4, 'b', 2, 5, 7, 8]* *steps: 19*	*[1, 2, 3, 4, 5, 'b', 7, 8, 6]* *[1, 2, 'b', 4, 5, 3, 7, 8, 6]* *[1, 'b', 2, 4, 5, 3, 7, 8, 6]* *['b', 1, 2, 4, 5, 3, 7, 8, 6]* *[4, 1, 2, 'b', 5, 3, 7, 8, 6]* *[4, 1, 2, 5, 'b', 3, 7, 8, 6]* *[4, 1, 2, 5, 3, 'b', 7, 8, 6]* *[4, 1, 2, 5, 3, 6, 7, 8, 'b']* *[4, 1, 2, 5, 3, 6, 7, 'b', 8]* *[4, 1, 2, 5, 3, 6, 'b', 7, 8]* *[4, 1, 2, 'b', 3, 6, 5, 7, 8]* *['b', 1, 2, 4, 3, 6, 5, 7, 8]* *[1, 'b', 2, 4, 3, 6, 5, 7, 8]* *[1, 3, 2, 4, 'b', 6, 5, 7, 8]* *[1, 3, 2, 4, 6, 'b', 5, 7, 8]* *[1, 3, 'b', 4, 6, 2, 5, 7, 8]* *[1, 'b', 3, 4, 6, 2, 5, 7, 8]* *[1, 6, 3, 4, 'b', 2, 5, 7, 8]* *steps: 19*

Initial State 4: [1, 2, 3, 4, 5, 6, 'b', 7, 8] *Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']*	*Initial State 4: [1, 2, 3, 4, 5, 6, 'b', 7, 8]* *Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']*	*Initial State 4: [1, 2, 3, 4, 5, 6, 'b', 7, 8]* *Solution Path: [1, 2, 3, 4, 5, 6, 7, 8, 'b']*
[1, 2, 3, 4, 5, 6, 7, 'b', 8] *[1, 2, 3, 4, 5, 6, 'b', 7, 8]* *steps: 3*	*[1, 2, 3, 4, 5, 6, 7, 'b', 8]* *[1, 2, 3, 4, 5, 6, 'b', 7, 8]* *steps: 3*	*[1, 2, 3, 4, 5, 6, 7, 'b', 8]* *[1, 2, 3, 4, 5, 6, 'b', 7, 8]* *steps: 3*
Initial State 5: [1, 2, 3, 4, 5, 6, 7, 'b', 8] *Solution Path: [1, 2, 3, 4, 5, 6, 7, 'b', 8]* *[1, 2, 3, 4, 5, 6, 7, 8, 'b']* *steps: 2*	*Initial State 5: [1, 2, 3, 4, 5, 6, 7, 'b', 8]* *Solution Path: [1, 2, 3, 4, 5, 6, 7, 'b', 8]* *[1, 2, 3, 4, 5, 6, 7, 8, 'b']* *steps: 2*	*Initial State 5: [1, 2, 3, 4, 5, 6, 7, 'b', 8]* *Solution Path: [1, 2, 3, 4, 5, 6, 7, 'b', 8]* *[1, 2, 3, 4, 5, 6, 7, 8, 'b']* *steps: 2*
Average number of steps: (2+15+21+19+3)/5 = 12	*Average number of steps: (2+15+21+19+3)/5 = 12*	*Average number of steps: (2+15+21+19+3)/5 = 12*

I have implemented Best-first and A* search algorithms with three heuristics, and observed that A* search has performed better than Best-first. Best-first has performed variably with different heuristics.

For BFS, f(n) only depends on h(n), thus the strength of each type of heuristic used is clearly showcased when using BFS. Therefore, selection of heuristic should be made with careful consideration. Considering the heuristics used , Manhattan distance works better and is followed by misplaced tiles. The heuristic which has responded well is best

suited for the problem.The ability to move a tile only up or down corresponds directly to manhattan distance.

For A*, consider both g(n) and h(n), g(n) i.e. the cost of reaching the current node from the root node and h(n) will balance each other and present the cheapest path to the goal. If we consider the three heuristics that I used for this problem, A* is the only algorithm that gives uniform results. Therefore A* is a better algorithm for the given problem.

15 puzzle problem:

For most of the relatively medium level inputs, the 15 puzzle code couldn't return a solution within the arbitrary number of steps.

Best-First Search:

Heuristic1: Manhattan Distance	*Heuristic2: Euclidean Distance*	*Heuristic3: Misplaced Tiles*
Initial Path1:[1, 2, 3, 'b', 5, 6, 8, 4, 9, 10, 7, 11, 13, 14, 15, 12] *Steps: 6*	*Initial Path1:[1, 2, 3, 'b', 5, 6, 8, 4, 9, 10, 7, 11, 13, 14, 15, 12]* *Steps: 6*	*Initial Path 1: [1, 2, 3, 'b', 5, 6, 8, 4, 9, 10, 7, 11, 13, 14, 15, 12]* *Steps: 6*
Initial Path 2: ['b', 6, 2, 3, 1, 5, 8, 4, 9, 11, 7, 12, 13, 10, 14, 15] *Steps: Unreachable*	*Initial Path 2: ['b', 6, 2, 3, 1, 5, 8, 4, 9, 11, 7, 12, 13, 10, 14, 15]* *Steps: Unreachable*	*Initial Path 2: ['b', 6, 2, 3, 1, 5, 8, 4, 9, 11, 7, 12, 13, 10, 14, 15]* *Steps: Unreachable*

Initial Path 3: [15, 2, 3, 'b', 5, 6, 8, 4, 9, 10, 7, 11, 13, 14, 1, 12] *Steps: Unreachable*	*Initial Path 3: [15, 2, 3, 'b', 5, 6, 8, 4, 9, 10, 7, 11, 13, 14, 1, 12]* *Steps: Unreachable*	*Initial Path 3: [15, 2, 3, 'b', 5, 6, 8, 4, 9, 10, 7, 11, 13, 14, 1, 12]* *Steps: Unreachable*
Initial Path 4: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 'b', 13, 14, 15] *Steps: 4*	*Initial Path 4: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 'b', 13, 14, 15]* *Steps: 4*	*Initial Path 4: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 'b', 13, 14, 15]* *Steps: 4*
Initial Path 5: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 'b'] *Steps: 2*	*Initial Path 5: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 'b']* *Steps: 2*	*Initial Path 5: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 'b']* *Steps: 2*
Average number of steps: 4	*Average number of steps: 4*	*Average number of steps: 4*

A* Search:

Heuristic1: Manhattan Distance	*Heuristic2: Euclidean Distance*	*Heuristic3: Misplaced Tiles*

Initial Path 1: [1, 2, 3, 'b', 5, 6, 8, 4, 9, 10, 7, 11, 13, 14, 15, 12] *Steps: 6*	*Initial Path 1: [1, 2, 3, 'b', 5, 6, 8, 4, 9, 10, 7, 11, 13, 14, 15, 12]* *Steps: 6*	*Initial Path 1: [1, 2, 3, 'b', 5, 6, 8, 4, 9, 10, 7, 11, 13, 14, 15, 12]* *Steps: 6*

Initial Path 2: ['b', 6, 2, 3, 1, 5, 8, 4, 9, 11, 7, 12, 13, 10, 14, 15] *Steps: Unreachable*	*Initial Path 2: ['b', 6, 2, 3, 1, 5, 8, 4, 9, 11, 7, 12, 13, 10, 14, 15]* *Steps: Unreachable*	*Initial Path 2: ['b', 6, 2, 3, 1, 5, 8, 4, 9, 11, 7, 12, 13, 10, 14, 15]* *Steps: Unreachable*
Initial Path 3: [15, 2, 3, 'b', 5, 6, 8, 4, 9, 10, 7, 11, 13, 14, 1, 12] *Steps: Unreachable*	*Initial Path 3: [15, 2, 3, 'b', 5, 6, 8, 4, 9, 10, 7, 11, 13, 14, 1, 12]* *Steps: Unreachable*	*Initial Path 3: [15, 2, 3, 'b', 5, 6, 8, 4, 9, 10, 7, 11, 13, 14, 1, 12]* *Steps: Unreachable*
Initial Path 4: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 'b', 13, 14, 15] *Steps: 4*	*Initial Path 4: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 'b', 13, 14, 15]* *Steps: 4*	*Initial Path 4: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 'b', 13, 14, 15]* *Steps: 4*

Initial Path 5: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 'b'] *Steps: 2*	*Initial Path 5: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 'b']* *Steps: 2*	*Initial Path 5: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 'b']* *Steps: 2*
Average number of steps: 4	*Average number of steps: 4*	*Average number of steps: 4*