CSC384 - Assignment 1: Advanced Heuristic

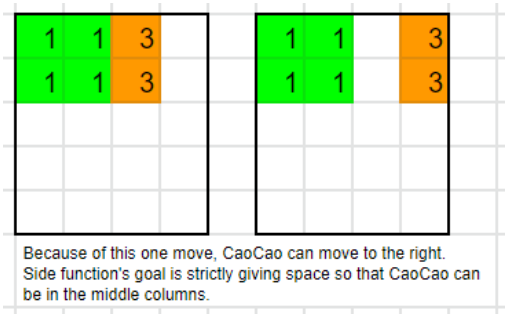
The advanced heuristic is built on top of Manhattan Distance, where the heuristic calculates an additional value to the Manhattan Distance value. It looks at CaoCao's neighboring tiles that is beside and/or below CaoCao. Manhattan Distance calculates the steps it takes for CaoCao to move to its goal position. This heuristic adds the steps/tile movements it takes for the neighboring tiles that prevent CaoCao to get to the goal position. This neighboring tile movement calculation dismisses the fact that there are other tiles apart from CaoCao and its adjacent tiles (neighbors). It can be said that the non-adjacent tiles are treated as blank tiles. The logic is as follows

- If CaoCao is at the bottom of the puzzle and it's not goal state, calculate the side neighbors' movements to clear the path for CaoCao to get to the goal state.
- Else if CaoCao is in the middle of the puzzle column-wise, calculate the bottom neighbors' movements to clear the path for CaoCao to get to the goal state.
- Else if CaoCao is at the leftmost column or rightmost column, calculate both bottom and side neighbors' required movements.

This bottom and side calculations can be calculated in functions. The bottom function will investigate the adjacent tiles below CaoCao and see how many steps each adjacent tile needs to move so that CaoCao can move (strictly) downwards. The side function will investigate the adjacent tiles beside CaoCao and see how many steps each adjacent tile needs to move so that CaoCao can move towards the middle. The additional tile movements depend on the adjacent tile type.

Neighbor direction	Tile type	Requires **at least** … tile movements	Note
Bottom	2 (2x1)	1	If CaoCao is at leftmost or rightmost column
		3	If CaoCao is at the middle column
	3 (1x2)	1	
	4 (1x1)	1	
Side	2	1	The side function looks at the direction towards the middle columns. If CaoCao is at leftmost column, CaoCao's right neighbors will undergo the calculations. If CaoCao is at rightmost, it will look at its left adjacent tiles.
	3		
	4		

Example of tile type 3 side move



This advanced heuristic dominates Manhattan since its result is always greater or equal Manhattan Distance. The advanced heuristic adds an additional value to the Manhattan Distance. It is also admissible since the heuristic will not be larger than the shortest path of the state to the goal node. For example, for Manhattan Distance of 1 states, there wouldn't be an additional number to the heuristic if there is no tile blocking CaoCao to the goal state. If two 1x1 tiles are in the way, the heuristic adds up to 3, which is the minimum steps it takes to get rid of the blocking tiles. This heuristic treats as though the blocking tiles can be moved anywhere in the puzzle (to a lot of blank tiles) for CaoCao to go to its goal state. This is the minimum steps! If there are other tiles blocking the two 1x1 tiles to clear out a way for CaoCao, that means the actual cost of cheapest path is larger than this.

To run the code: A_Search(infilepath, outfilepath, "Advanced")