CS 4200 - Project 1

Noah Reef

Winter 2023

1 Code Outline

For this project, I implemented three main functions: `A_star`, `manhattan_distances`, and `displaced_distances`. The `A_star` method implements A^*-search using *graph search* utilizing both a frontier and explored set to keep track of visited states and states to be explored next. `manhattan_distances` and `displaced_distances` compute the Manhattan distance and number of displaced tiles heuristics, respectively.

For the purpose of the project the 8-puzzles are represented using `numpy`'s `np.matrix`, hence each puzzle state is viewed as a 3×3 matrix. This representation made computing Manhattan distances simply the difference in element index locations within the matrix.

Some auxiliary functions are `read_file` which read in input matrix files, `Experiment` which conducted the experiment whose results are seen in the Section 2, and `check_is_valid` which checks if a given puzzle state is solvable.

2 Experiment Results

I ran ran A^*-search on the provided puzzles in *Length4.txt*,*Length8.txt*,*Length12.txt*, *Length16.txt*, and *Length20.txt* respectively. I recorded the search cost (i.e. the number of nodes explored) for both the Manhattan Distance and Displaced Tile Heuristic with the time to solve recorded for each method. The results are shown below,

d	Manhattan Distance	Manhattan Distance Time	Displaced Tiles	Displaced Tiles Time
4	4	0.0007317066192626950	4	0.0003480911254882810
4	4	0.0004827976226806640	4	0.0004160404205322270
4	4	0.0004441738128662110	4	0.000410795211791992
4	4	0.0004389286041259770	4	0.000405073165893555
4	4	0.0004401206970214840	4	0.000402927398681641
4	4	0.0004327297210693360	4	0.000390291213989258
4	4	0.0004317760467529300	4	0.000391006469726563
4	4	0.0004332065582275390	4	0.000526666641235352
4	4	0.0004396438598632810	4	0.000395059585571289
4	4	0.0004479885101318360	4	0.000401258468627930

Figure 1: A^*-Search on $d = 4$

d	Manhattan Distance	Manhattan Distance Time	Displaced Tiles	Displaced Tiles Time
8	8	0.0008640289306640630	8	0.0008189678192138670
8	11	0.0013499259948730500	11	0.0034132003784179700
8	10	0.0011892318725585900	10	0.001912832260131840
8	9	0.001032114028930660	9	0.0017828941345214800
8	11	0.0012791156768798800	11	0.001920938491821290
8	12	0.0014081001281738300	12	0.0037009716033935500
8	9	0.0010433197021484400	9	0.0021677017211914100
8	4	0.0004360675811767580	4	0.0005099773406982420
8	10	0.001177072525024410	10	0.0019059181213378900
8	12	0.0014400482177734400	12	0.0037298202514648400

Figure 2: A^*-Search on $d = 8$

d	Manhattan Distance	Manhattan Distance Time	Displaced Tiles	Displaced Tiles Time
12	27	0.003608226776123050	27	0.01557302474975590
12	16	0.002048015594482420	16	0.020626068115234400
12	14	0.0016400814056396500	14	0.01711893081665040
12	19	0.002340078353881840	19	0.021529674530029300
12	53	0.009799003601074220	53	0.03615403175354000
12	16	0.0018620491027832000	16	0.016961097717285200
12	18	0.002223968505859380	18	0.020889997482299800
12	6	0.0006458759307861330	6	0.0006153583526611330
12	37	0.005861759185791020	37	0.03341317176818850
12	47	0.00810098648071289	47	0.0294189453125

Figure 3: A^*-Search on $d = 12$

d	Manhattan Distance	Manhattan Distance Time	Displaced Tiles	Displaced Tiles Time
16	43	0.006573915481567380	43	0.38930511474609400
16	59	0.011310100555419900	59	0.47799205780029300
16	43	0.0068018436431884800	43	0.45833802223205600
16	42	0.007061958312988280	42	0.136397123336792
16	178	0.06787300109863280	178	0.9285287857055660
16	27	0.0037262439727783200	27	0.38691282272338900
16	20	0.0025289058685302700	20	0.07240509986877440
16	11	0.0012710094451904300	11	0.004542827606201170
16	101	0.026643991470336900	101	0.7700259685516360
16	102	0.028192996978759800	102	0.7773878574371340

Figure 4: A^*-Search on $d = 16$

d	Manhattan Distance	Manhattan Distance Time	Displaced Tiles	Displaced Tiles Time
20	165	0.053826093673706100	165	22.081565856933600
20	460	0.3998410701751710	460	17.709371328353900
20	394	0.2928287982940670	394	16.735992193222000
20	80	0.017946958541870100	80	3.7356631755828900
20	458	0.37535691261291500	458	18.883805751800500
20	178	0.06432485580444340	178	20.904850006103500
20	58	0.011627912521362300	58	2.0911049842834500
20	54	0.010071992874145500	54	0.15242695808410600
20	297	0.1689159870147710	297	15.46637511253360
20	205	0.09166407585144040	205	13.536034107208300

Figure 5: A^*-Search on $d = 20$

here is a summary of the results shown below,

Length	Avg Manhattan Distance	Avg Manhattan Distance Time	Avg Displaced Tiles	Avg Displaced Tiles Time
4	4.0	0.0004723072052001950	4.1	0.0004087209701538090
8	9.6	0.0011219024658203100	16.7	0.0021863222122192400
12	25.3	0.003813004493713380	81.9	0.021230030059814500
16	62.6	0.016198396682739300	446.2	0.44018356800079300
20	234.9	0.14864046573638900	2605.0	13.129718947410600

Figure 6: Summary of Results

3 Analysis

Here we see that roughly the time grows exponentially with the growth in solution depth for both heuristics, however using the *Displaced Tiles* heuristic grew at a faster rate for our A^*-search than the Manhattan Distance heuristic. This difference in time is also reflected in the search cost of both approaches, in which on average Manhattan Distance finds its solution with a smaller search cost than Displaced Tiles, indicating that Manhattan Distance is a better heuristic for the 8-puzzle problem than Displaced Tiles.