CSE3521 SP21
John Choi.1655
Austin Schall.37

2.

Initial State:

1	**0**	**3**
5	**2**	**6**
4	**8**	**7**

Goal State:

1	**2**	**3**
8	**0**	**4**
7	**6**	**5**

Initial State:

1	**2**	**3**
4	**5**	**0**
8	**7**	**6**

Goal State:

1	**2**	**3**
8	**0**	**4**
7	**6**	**5**

Initial State:

0	**1**	**3**
4	**2**	**6**
8	**5**	**7**

Goal State:

1	**2**	**3**
8	**0**	**4**
7	**6**	**5**

Do the results from functions match expectations? Both functions, is_goal_state() and find_successors(), ran just like they should. is_goal_state() checked to see if the tables were in the proper goal state and returns either that it's not or that it is. find_successors() reports back the possible steps you can take, which ended up working just fine.

7.

Initial State: **1** | **0** | **3** **5** | **2** | **6** **4** | **8** | **7**	Results: **EVALUATED/EXPANDED/LENGTH** BFS: 87829 / 87829 / 73221 DLS 1: 2461972 / 895265 / 14 DLS 2: 2461986 / 895279 / 28 IDS: 1537243 / 537787 / 14 A*: 58 / 58 / 58
1 | **2** | **3** **4** | **5** | **0** **8** | **7** | **6**	Results: **EVALUATED/EXPANDED/LENGTH** BFS: 132893 / 132893 / 89579 DLS 1: 3200 / 1165 / 8 DLS 2: 22517 / 8194 / 16 IDS: 2415 / 847 / 8 A*: 18 / 18 / 18
0 | **1** | **3** **4** | **2** | **6** **8** | **5** | **7**	Results: **EVALUATED/EXPANDED/LENGTH** BFS: 29594 / 29594 / 28000 DLS 1: 402269 / 146282 / 13 DLS 2: 146297 / 50301 / 27 IDS: 306777 / 107796 / 13 A*: 55 / 55 / 55
Explain: As expected, A* algorithm had the fastest runtime for all boards. BFS was the slowest out of the 4 algorithms that we have, and DLS had the shortest path to the goal.	

8.

<u>**Initial State:**</u>

1	**3**	**0**
5	**2**	**6**
4	**8**	**7**

Results:
EVALUATED/EXPANDED
BFS: 87827 / 87827
DLS 1: 2461973 / 895266
DLS 2: 895283 / 307767
IDS: 2179235 / 767838
A* Manhatten: 59 / 59
A* Misplaced Tile: With our current solution, this heuristic value would crash because it reaches the point where all the current node's children are already visited but no goal has been found. This would create a loop.

A* "stupid" heuristic: With our current solution, this heuristic value would crash because it reaches the point where all the current node's children are already visited but no goal has been found. This would create a loop.