cmput 355 2022 assignment 3

Do not submit any answers to this assignment: every registered student will automatically receive full marks. :) Solutions will be posted soon.

1. For each of the two positions below, draw the next two levels of the sliding tile search space graph.

```
1 2 3          1 2 4
4 5 6          3 5 6
7 8 _          7 8 _
```

   If you continued to draw each of these two components, level by level, you would see that the two components are isomorphic (have the same shape). Why is that?

2. You can use `stile_search_v2.py` to find a hardest (solution is the longest) 3x3 sliding tile puzzle.

   step 0) execute `git pull` in your copy of the class github repo to make sure that you have the latest version.

   step 1) On the target position T, switch the labels on any two tiles with consecutive numbers. E.g. below I have switched the labels on the tiles with numbers 3,4. Call the new position Q. Explain why the number of inversions in Q is exactly 1 and why Q is unsolvable.

```
1 2 3          1 2 4
4 5 6          3 5 6
7 8 _          7 8 _
```

   step 2) Run `stile_search_v2.py` on Q. It will report that Q is unsolvable, and also print out the last position (call it Z) encountered in the bfs that started at Q. What position is Z? In the sliding tile search space component that starts with Q (so Q will be at level 0 in that component), at what level is Z?

   step 3) Let Y be the position you get by taking Z and switching the same two labels you switched in step 1. What position is Y? Is it solvable or unsolvable? How many moves does it take to solve Y How do you know every 3x3 sliding tile puzzle can be solved in this many moves or fewer?

3. (i) Three sliding tile heuristics — inversions, misplaced tiles, manhattan distance — are commonly used to estimate the number of moves to the goal. Which one is used in the above execution? How can you tell?

   (ii) Will the above execution eventually find a shortest solution? Explain briefly.

4.

```
_ 2        5 2
5 3        _ 3
4 1        4 1
```

Start with the 3×2 sliding tile position above left and slide tile 5 up. The change in the number of inversions is +1: before, 2 and 5 were not inverted; after, they are inverted; every other tile pair has its inversion status unchanged.

Assume that you have a sliding tile position with dimensions 5x3 and with the blank space in row 2 and column 2. Assume that you move the tile below the blank up. Give every possible change in the number of inversions. (E.g. answer $\{+1, 0, -2\}$ means that you think that the number of inversions can sometimes increase by one, sometimes be unchanged, and sometimes decrease by 2, and no other change is possible.) Justify your answer.

5. The *transpose* of an array is the array you get by interchanging rows and columns. E.g. below left are two arrays, at right are their transposes.

```
4 8   1 3 5          4 1 7   1 2 _
1 _   2 4 6          8 _ 5   3 4 7
7 5   _ 7 8                  5 6 8
```

```
4 8 2
1 _ 3
7 5 6
```

For the position above, give a solution by drawing the sequence of positions that leads to the solution position. For each position in this sequence, give the number of inversions, the number of transpose inversions (inversions of the transpose position), and the taxicab distance. Format your answer as shown below.

```
4 8 2  . . .  . . .  . . .  . . .  . . .  . . .  . . .  . . .  . . .  . . .  1 2 3
1 _ 3  . . .  . . .  . . .  . . .  . . .  . . .  . . .  . . .  . . .  . . .  4 5 6
7 5 6  . . .  . . .  . . .  . . .  . . .  . . .  . . .  . . .  . . .  . . .  7 8 _
```

```
inv :  ?    ?    ?    ?    ?    ?    ?    ?    ?    ?    0
tinv:  ?    ?    ?    ?    ?    ?    ?    ?    ?    ?    9
taxi:  ?    ?    ?    ?    ?    ?    ?    ?    ?    ?    0
```

6. For a sliding tile position p, define I(p), T(p), X(p) to be respectively p's number of inversions, number of transpose inversions, and taxicab distance to the solution. Which of these functions do you think would be a best choice for the heuristic of an A* implementation of a sliding tile solver? Justify briefly.

I(p) T(p) X(p) (I(p)+T(p))/2.0 (I(p)+T(p)+X(p))/3.0

7. Do the next step in the course webpage example of A* finding a path from A to B. Format your answer exactly as on the webpage, as started below.

```
R nbrs:
C ...
P ...
S ...
      S    T    Z    F    O    R    ...
cost .....
heur .....
pri ......
```

8. Here is the start of an execution of A* on the sliding tile puzzle. MSF is moves so far. EMTG is estimated moves to the goal. Show step 6. (There might be more than one correct answer.) Explain your work.

```
step   current position     positions added to queue
 1.    p0  3 6 7  msf:0     p1  3 6 7  msf:1     p2  3 6 7  msf:1
           2 5 4  emtg:7        2 5 4  emtg:7        - 5 4  emtg:7
           - 1 8                1 - 8                2 1 8
                               in queue  - 1 2 - - - - - - -

 2.    p2  3 6 7  msf:1     p3  - 6 7  msf:2     p4  3 6 7  msf:2
           - 5 4  emtg:7        3 5 4  emtg:7        5 - 4  emtg:8
           2 1 8                2 1 8                2 1 8
                               in queue  - 1 - 3 4 - - - - -

 3.    p1  3 6 7  msf:1     p5  3 6 7  msf:2     p6  3 6 7  msf:2
           2 5 4  emtg:7        2 5 4  emtg:6        2 - 4  emtg:8
           1 - 8                1 8 -                1 5 8
                               in queue  - - - 3 4 5 6 - - -

 4.    p5  3 6 7  msf:2     p7  3 6 7  msf:3
           2 5 4  emtg:6        2 5 -  emtg:6
           1 8 -                1 8 4
                               in queue  - - - 3 4 - 6 7 - -

 5.    p7  3 6 7  msf:3     p8  3 6 7  msf:4     p9  3 6 -  msf:4
           2 5 -  emtg:6        2 - 5  emtg:7        2 5 7  emtg:6
           1 8 4                1 8 4                1 8 4
                               in queue  - - - 3 4 - 6 - 8 9
```