**ABHINAV SHARMA – 4 CREDITS**

**SHYAM RAJENDRAN - 4 CREDITS**

**SAIKAT ROYCHOWDHURY - 4 CREDITS**

We have implemented the following enhancements in code for faster solution convergence.

1. Since a given board instance heuristic is constant, we pre compute the heuristic value for each instance of the board during its initialization (constructor) using a map and bring the score lookup time to O(1).
2. We have used a one dimensional representation of 8 puzzle. This helps us avoid any costly one-one matching with a goal board. (board\_array[i] = i

is the invariant for our goal board). Also such a layout allows an easy offset based access of the board tiles. This also means space efficiency, since we do not explicitly store any goal board for each board instance.

Gashnig’s Heuristic –

We compute Gashnig’s heuristic score as follows

1. CASE I - Blank cell is at correct position.

We maintain a list of misplaced tiles in a list. The list is sorted by tile number i.e a misplaced tile with lower id is placed before a higher misplaced tile. We peek at the head of the sorted list this gives the tile and its current position in the 8 puzzle. We swap the tile with the blank space and update the new position of the tile.

1. CASE II - Blank cell is not at correct position

For this scenario, we determine which tile is the correct owner of the blank cell. Then that tile is located and replaced with this blank cell. Thus the swapped tile has now reached its correct position in the 8 puzzle. So we remove it off the list of misplaced tiles.