COMS 4720: Project 1

Jacob Duba

October 2024

Let s be a state and m be the Manhattan distance heuristic. My heuristic for when double moves are allowed can be described as such:

$$h_3 = \left\lceil \frac{m(s)}{2} \right\rceil \tag{1}$$

My intuition behind this is that when double moves are used, the cost of moving a single tile is 0.5. For example, the cost for each of the tiles moved in the example below is 0.5:

```
1 2 3  DBL_LEFT  1 2 3
4 5 6 ---------> 4 5 6
0 7 8            7 8 0
```

The total cost was 1 and can be divided between tiles 7 and 8. To keep the heuristic admissible, I acknowledge that a best-case scenario could only use double moves. So, the heuristic divides the Manhattan distances by 2. However, there is one situation in which a single move will be guaranteed to be used: when the Manhattan distance is odd. You can only move an even number of tiles 2 at a time, so the last tile will be a guaranteed single move. That is why the heuristic rounds up.

Here's my code:

```
private int computeNumSingleDoubleMoves() {
    if (numSingleDoubleMoves > -1) {
        return numSingleDoubleMoves;
    }

    numSingleDoubleMoves = (computeManhattanDistance() + 1) / 2;

    return numSingleDoubleMoves;
}
```