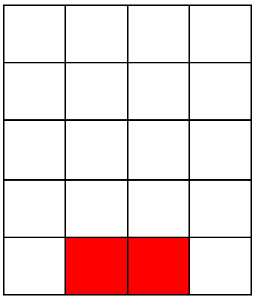
This is my function to calculate heuristics

```
h = abs(3-i) + abs(1-ii) + (int((state[4][1] == 2) or (state[4][1] == 3)) +
int((state[4][2] == 2) or (state[4][2] == 3)))*0.5
```

It’s just Manhattan distance function but I add one more number. This number changes depending on what stays on two places in the lower bound:



If these places are occupied by the 2s(wide tile) or 3s(long tile) we add 0.5 to heuristics function for each place. This heuristics function is admissible. H(n) for the winning state is 0. My heuristics function dominates Manhattan distance because one of the addends is Manhattan distance and my second addend may be only 0, 0.5 or 1. And in some states overall heuristics is strictly more than Manhattan distance:

3	1	1	3
3	1	1	3
3	0	4	4
3	0	3	4
2	2	3	4

In this situation Manhattan distance is 3 and my heuristics function equals to 4.