

## Conclusions from the A\* Search algorithm

### The zero “heuristic”

When running 100 runs of 3x3 sliding puzzles using the “zero heuristic”. This is essentially the uniform cost algorithm. This took a similar amount of time (if not slightly longer) to the iterative deepening search from last exercise. The average time per solution was **0.084s**.

Furthermore, there were on average over 7000 states pushed onto the frontier before the solution was found.

When trying the same with 4x4 puzzles, it took far too long to even do one puzzle, so I moved on to the next test.

### The number of tiles out of place

When replacing the heuristic function with one that counts the number of misplaced tiles, the average time per solution was **0.0037s**. That’s over **twenty** times faster. The average number of states inserted into the frontier was also reduced to about 200.

With 4x4 puzzles, this heuristic managed to do some puzzles very fast, but quickly got stuck on some others. Therefore, there is no average running time for these problems.

### Manhattan distance

Using the total manhattan distance of all the tiles as a heuristic obtained an astounding **0.0018s**. That’s double the speed of the previous attempt. This time, only 57 states on average were inserted into the frontier.

With 4x4 puzzles, it occasionally got stuck but only for a couple of seconds at a time before continuing a burst of rapid solutions. Overall the average time per run was **0.69s** with an average of 13,500 inserts.

These results were so fast, that I was able to run a few 5x5 puzzles with an average time of 10.6s before one got stuck for a minute and I got bored of waiting.