Mandatory Project 1

Computationally hard problems

October 27, 2015

Troels Thomsen 152165

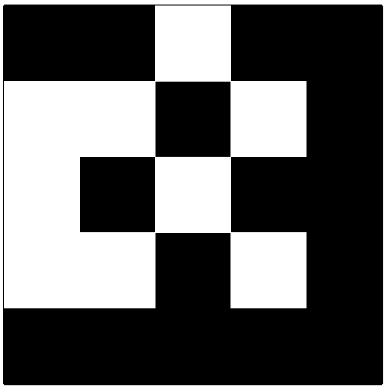
Rasmus Haarslev 152175

1 Madragon

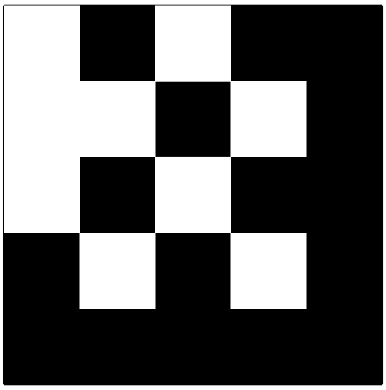
1.1 Determine whether the answer to the given Madragon instance is YES or NO

We take the following steps to shift the starting board to the goal board.

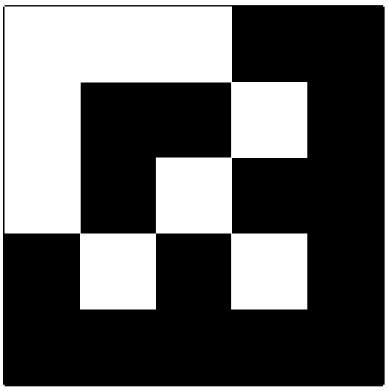
Starting board:



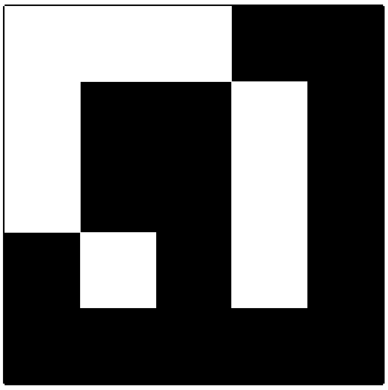
We move column 1 one tile up.



We column 2 two tiles down.



We move row 3 two tiles to the left.



And now we have made $k = 3$ operations and achieved the goal.

1.2 Show that Madragon is in NP

We specify the following randomized deterministic polynomial-bound algorithm A, which takes as input a problem instance X containing board states a and b and a random sequence of integers R.

1. 1a R is a random sequence of k integers chosen from an even distribution of numbers ranging from 1 to $(m+n)^2$.

   1b Every index in R corresponds a step. The value of each index corresponds to a move of either a column or a row. Since there exists $(m+n)^2$ unique moves, the value of R_i will determine which of these moves the algorithm performs at the ith step.

   1c After A has performed all i steps in R, A checks if the board has reached its goal state b and answers YES or NO accordingly.

2. 2a Since we choose R from an evenly distributed set of all possible combinations of size $\{1, \cdots, (m+n)^2\}^k = (m+n)^{2k}$, we know that if at least one R exists which solves $A(X, R) = YES$, we can choose this R with a probability of $\frac{1}{(m+n)^{2k}} > 0$.

   2b If there does not exist an R which solves $A(X, R) = YES$, then the algorithm A will never answer YES since the board cannot arrive at goal board b within k steps given any R.

3. In order to interpret R the algorithm A must generate all possible moves for the given board. This can be done in $O((m+n)^2)$ time. We can perform all steps in R in $O(k)$ time, and check the solution in $O(1)$ time. Thus the algorithm is bound by $O((m+n)^2)$.

1.3 Show that Madragon is NP-complete