SDSU | San Diego State University

Optimizing 2048: Analyzing and Comparing Solving Algorithms for Improved Performance

Stamatis Frangoulis

sfrangoulis7616@sdsu.edu

1. Introduction

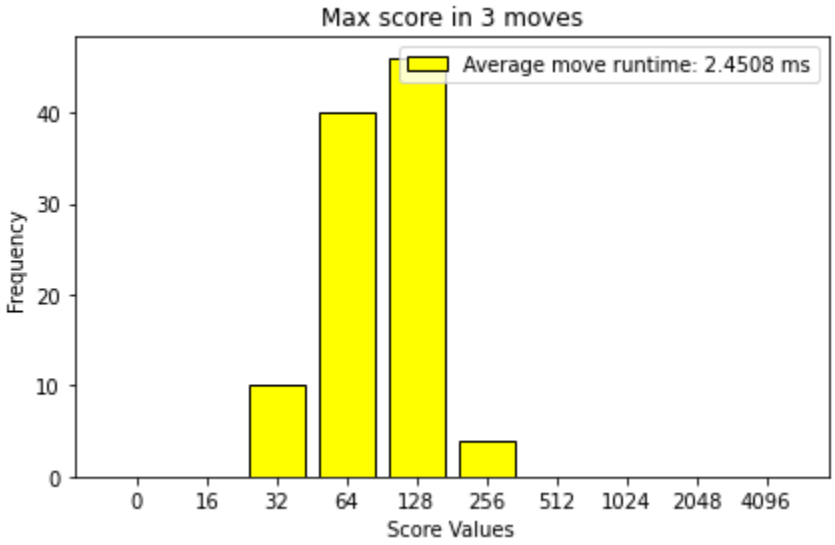
2048 is a simple game that was created by Gabrielle Cirulli in 2014. It involves 16 tiles with powers of two that can be moved up, down, left and right to merge into greater numbers. The aim of the game is to get 2048 into a tile however it is also possible to get to 4096, 8192 and so on. There is one overlying idea on how to get the best score by organizing the highest tiles towards a corner however when playing there can be various approaches. In this project, three methods analyze their mechanics, strengths, and drawbacks.

The concept of the 2048 game was replicated in code, allowing for the development and implementation of the solving algorithms. The initial code was developed in java using Intellij which allowed for the improvement of the game and the implication of solving algorithms. These were run and yielded some initial data that were visualized in excel. Following this, to improve efficiency the code was translated into python. Converting into python enabled the improvement of the code's efficiency and accuracy by reducing the runtime significantly. Furthermore, the python matplotlib library was beneficial in plotting the results of the various solving methods directly. This allows an easier visualization of each method's distribution of scores in the game but also their runtime before every move to evaluate the algorithm's efficiency. This yielded some interesting figures that are presented below.

2. Results

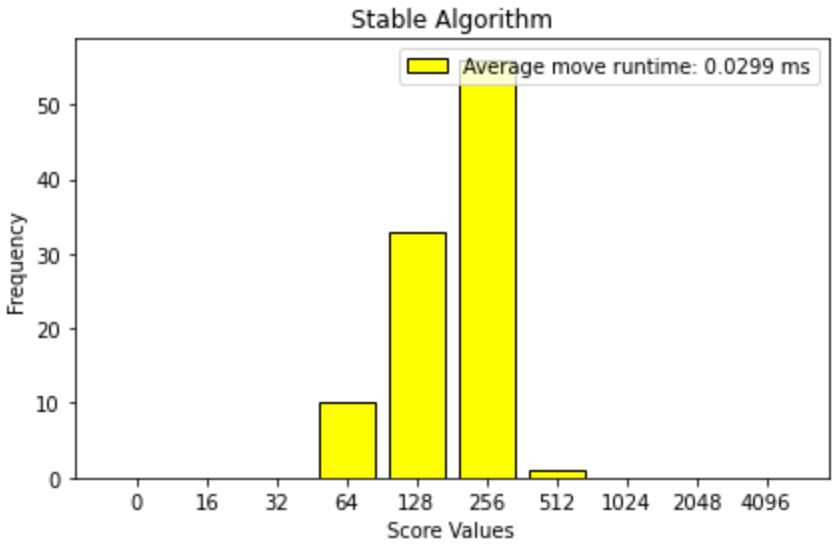
2.1. Maximum Score

This method analyzes all possible moves for a certain depth and picks the move that leads to the highest score. In the figure below the method was run 100 times with a depth of 3.



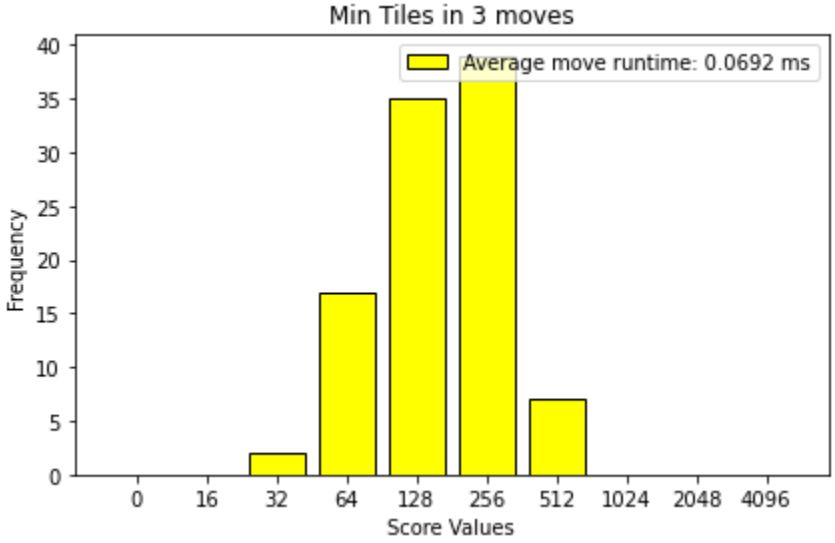
2.2. Stable Algorithm

This method follows the same pattern of first moving down, if not possible left, then right and lastly up. This code was also executed 100 times and yielded the following results.



2.3. Minimum Tiles

The minimum tiles method analyzes the next possible moves similarly to the maximum score method but chooses the move that will lead to the least amount of non-zero tiles. This method was run 100 times with a depth of 3 and generated the following results.



3. Conclusions

Overall, the Minimum Tiles method delivers the best performance, with scores typically falling between 512 and 1024. While its runtime before each move is comparable to the Maximum Score method, the Minimum Tiles approach outperforms in terms of score. In contrast, the Stable algorithm has a significantly shorter runtime than the Maximum Score method and achieves slightly better scores. While the Minimum Tiles method generally outperforms the Stable algorithm in terms of scoring, it suffers from a considerably longer runtime that increases factorially with each additional depth of move analysis. This highlights the potential benefits of combining the Stable algorithm with the Minimum Tiles approach, as it would reduce runtime and allow for deeper move analysis, ultimately leading to better performance.