

8-Parzzle Solver

Ahmad Rawwagah (afr2151), Kimberly Tsao (kt2803),
Ruth Lee (rsl2159)

November 27, 2023

1 Introduction

We plan to parallelize the 8-puzzle game. The 8-puzzle game is a game board with 8 movable tiles and one empty space. There is one tile for each number in the set 1, 2, 3, ... 8. The aim of this game is to get from any initial board state to the configuration with all tiles in ascending order from left to right, top to bottom. An example initial and ending state is as follows.

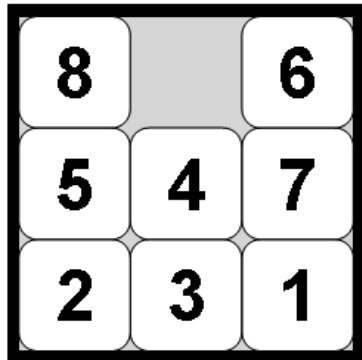


Figure 1: Initial State

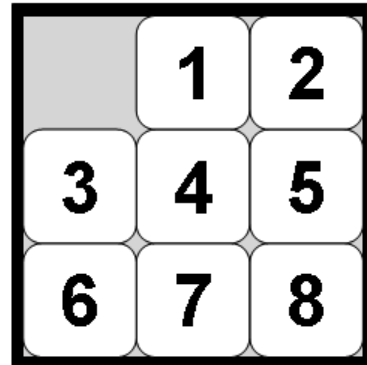


Figure 2: Goal State

2 Implementation

8-puzzle can be solved using different search algorithms. We will use BFS, DFS, and A* search algorithms and first write a sequential solver in Python. Then, we plan to parallelize this in a Haskell program that takes an initial state represented by a list of values 0-8 with 0 indicating the empty square. It will return an array of steps representing directions of "movement" for the empty square since is the easiest way to track tile movement.

We see opportunity to parallelize the heuristic function that A* search uses as well as using parMap to parallelize the search processes, breaking down the search space and distributing it across multiple threads. If time allows, we can also handle unsolvable boards by first counting the number of inversions in the input state as an odd number of inversions is unsolvable.

3 Evaluation

We will run time trials on different sequential implementations of the 8 Puzzle algorithm (BFS, DFS, A*) with our parallel implementation running on 2-48 threads. We will also evaluate the number of sparks created and converted in order to determine an ideal thread count and minimize excessive spark creation. Ideally we hope to greatly improve the performance of the algorithm as it is a slow sequential algorithm. If we have time we will also evaluate N sized puzzles.

4 References

1. <https://cse.buffalo.edu/faculty/miller/Courses/CSE633/Weijin-Zhu-Spring-2020.pdf>
2. <https://doi.org/10.1002/int.10027>
3. <https://www.geeksforgeeks.org/check-instance-8-puzzle-solvable/>