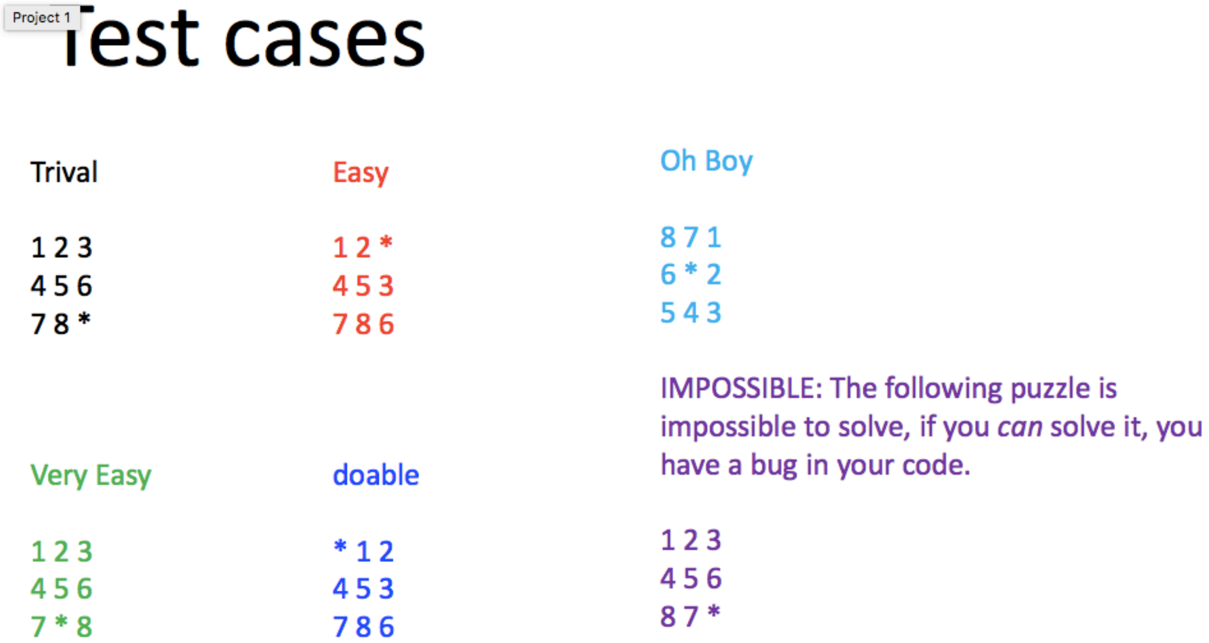
Items you can include in your report:

- Your challenges in this project
- Your design (objects and methods)
- Did you try optimizing your code by using special data structures that make searching for and comparing states faster?
- Did you implement a graph search (keeping record of all explored nodes), or did you use a tree search algorithm? If you tried both, you can compare the number of nodes created, and number of nodes expanded for each version (tree search and graph search). Note that you don’t ned to implement and compare both tree and graph searches. But if you did, you can write about your observations.
- Comparing Heuristic Functions in terms of time and space and analyzing the results with diagrams and/or tables. Sample tables and diagrams follow.
  - Your findings.
    Hint: Your findings for this project will be something like this:
    - For shallow problems, it does not matter too much what heuristic you use, if any.
    - As the problems get harder, heuristics help more and more.
    - A good heuristic is better than a weak heuristic.

Some test cases:



Project 1

Test cases

Trival

1 2 3
4 5 6
7 8 *

Easy

1 2 *
4 5 3
7 8 6

Oh Boy

8 7 1
6 * 2
5 4 3

IMPOSSIBLE: The following puzzle is impossible to solve, if you *can* solve it, you have a bug in your code.

1 2 3
4 5 6
8 7 *

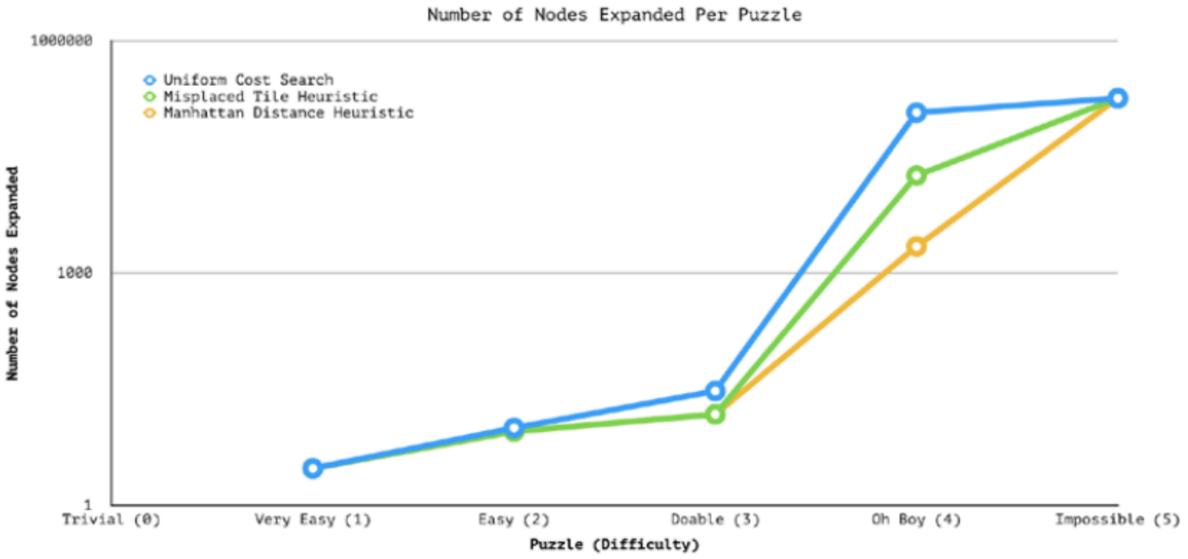
Very Easy

1 2 3
4 5 6
7 * 8

doable

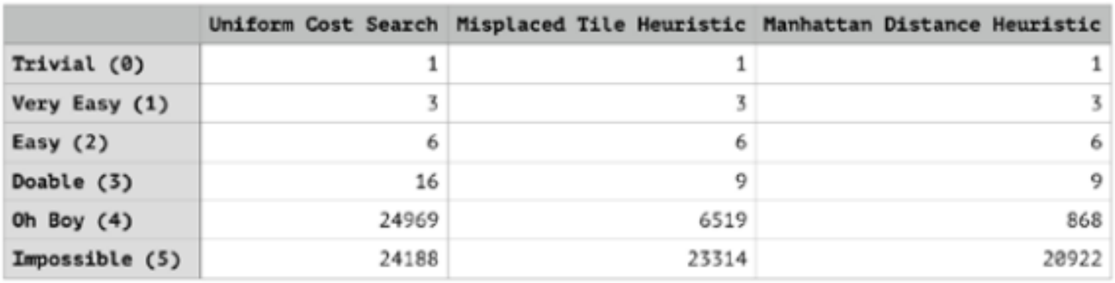
* 1 2
4 5 3
7 8 6

Sample diagrams and tables:

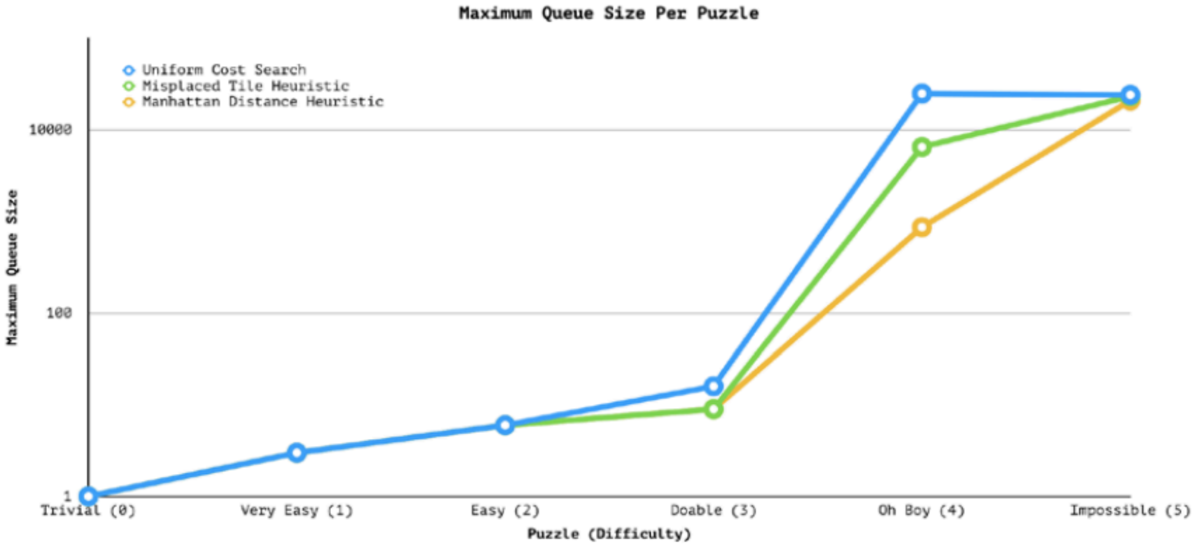


Number of Nodes Expanded Per Puzzle

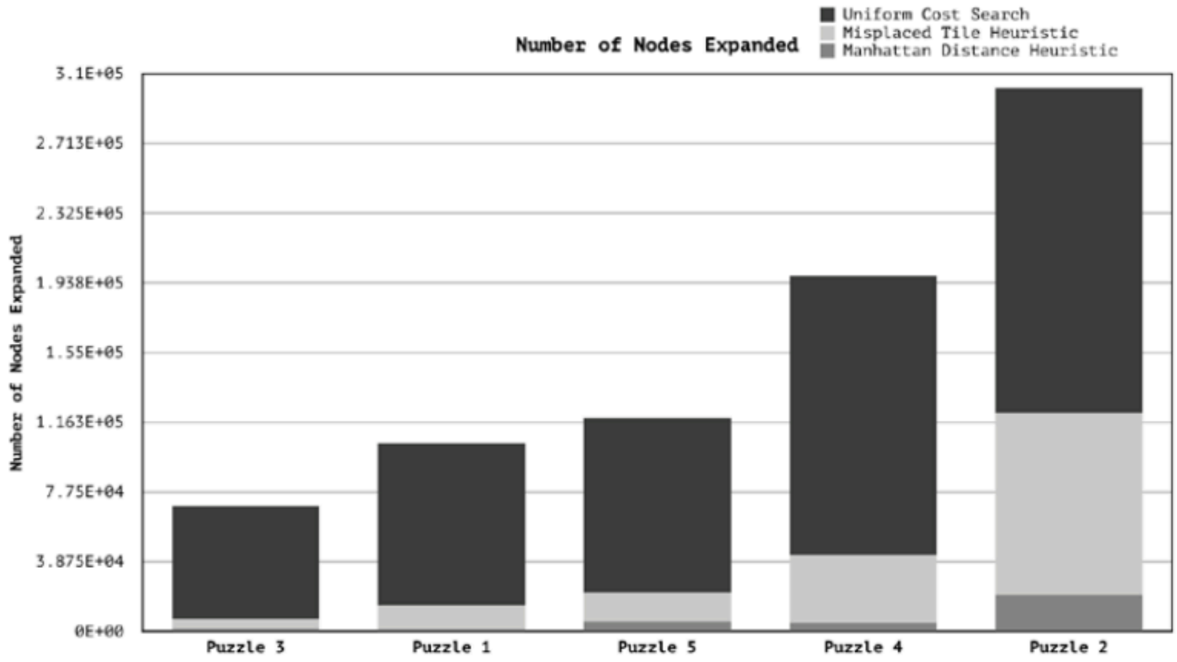
Maximum Queue Size



	Uniform Cost Search	Misplaced Tile Heuristic	Manhattan Distance Heuristic
Trivial (0)	1	1	1
Very Easy (1)	3	3	3
Easy (2)	6	6	6
Doable (3)	16	9	9
Oh Boy (4)	24969	6519	868
Impossible (5)	24188	23314	20922



Maximum Queue Size Per Puzzle



Number of Nodes Expanded

	Manhattan Distance Heuristic	Misplaced Tile Heuristic	Uniform Cost Search
Puzzle 3	1507	5340	62698
Puzzle 1	1144	13307	90374
Puzzle 5	5553	16166	96901
Puzzle 4	4608	38054	154909
Puzzle 2	20317	100978	180949