

CMPT225, Spring 2023

Final Project

Instructions:

You may do the project alone or in pairs. If you do the project in pairs you should write in the documentation who did each part. (It's ok if some parts were done together.)

You should submit a zip file called `fifteenpuzzle.zip` to Coursys.

The zip file will contain your **solution** and **documentation** of your project.

The solution should contain the class **fifteenpuzzle.Solver**. The `main()` method of **Solver** will take command line arguments with the names of the input file and the file that will contain a solution. That is, we invoke the solver from command line as follows:

```
java fifteenpuzzle.Solver input.txt sol.txt
```

Place all your files in the zip file under the folder `src`. That is, your zip file must contain at least one file: **src/fifteenpuzzle/Solver.java**. Your solution will most probably have more files.

Submit your solutions to Coursys before April 11, 23:59. No late submissions.

Project Description:

In this project you will write a solver for the [15 puzzle](#)

Your input will be a file containing the puzzle in the same format as in Assignment 1. You will need to solve the puzzle, and write the solution to the output file.

Several input files are provided with the assignment.

More inputs will be provided on March 27, and they will be used for grading.

If your algorithm doesn't find a solution to a specific puzzle for a long time, the TAs will skip it and move on to the next puzzle. Examples of possible outputs are provided.

To solve the puzzles you may use the graph exploration algorithms we discussed in class: BFS/DFS/A*. For A* you will need to think of different heuristics how to evaluate the distance from the current state to a solution.

*Some puzzles are hard, and you can get 100% on the project even if you don't solve everything.

Documentation:

I don't expect any specific format for documentation, you may use word/latex/plain text.

- Use plain English or diagrams or whatever they feel is right.
- Explain your choices of classes, data structures, algorithms, heuristics, etc.
- Was one of the heuristics always better? Or different heuristics solved different puzzles?
- Describe which parts were easier, which parts required more time.
- Describe what you wrote but then didn't include in the final project (e.g., switched from one data structure to another)

Good luck!