

CMPT225, Spring 2025  
Final Project  
Due date: December 2, 2025, 23:59. No late submissions.

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.)

**Project Description:** You will write a solver for Rubik's cube. Your input will be a file containing the scrambled cube in the same format as in Assignment 1. You will need to solve the scrambled cube, and write the solution to the output file. The output file is expected to have one line consisting of letters ***FBRLUD***, representing the sequence of rotations solving the scrambled cube.

**Grading:** Ten input files are provided with the assignment (and three examples of solutions). More inputs will be provided on November 25, and they will be used for grading.

If your algorithm doesn't find a solution to a specific input for more than 10 seconds, the TAs will skip it and move on to the next input. Examples of possible outputs are provided.

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

\* You can get 100% even if you don't find the shortest solution.

\* Some inputs might be hard. You can get 100% on the project even if you don't solve everything.



**Documentation:** In general, I don't expect any specific format for documentation.

- Submit the documentation in a pdf file.
  - Please write your names on top of the document
  - You can use plain English, diagrams, pictures of cats or whatever you feel is right.
  - Explain your choices of classes, data structures, algorithms, heuristics, etc.
  - Was one of the heuristics always better than another?
  - Describe which parts were easier, and which parts required more time.
  - Describe what you wrote, but did not include in the final project (e.g., switched from one data structure to another)
- 

**Submitting your project:** You need to submit a zip file called rubikscube.zip to Coursys. The zip file will contain your **solution** and **documentation** of your project.

The solution should contain the class **rubikscube.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 rubikscube.Solver scramble.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/rubikscube/Solver.java**. Your solution will most probably have more files.

**Good luck!**