Artificial Intelligence
Projects 1

Overview

Files: The files are available in the class git repo. The the course page for instructions on access the git server. The name of this assignment is `contest1`. You should only MySolver.h, MySolver.cpp and description.tex.

Code: The file code.pdf has information about the source code.

Grading: Each assignment will be graded as follows:

50%	Performance on test problem instances relative to reference implementations
20%	Code quality
20%	Description of algorithm used and why it is effective
10%	Phase 1 implementation of depth first search
up to 5% bonus	Performance relative to your classmates

Phase 1

Due: 12pm Thursday, September 13. Late assignments will not be accepted.

Goal: Write an agent to solve the eight puzzle using **depth first search** in an efficient manner. Do not try to do anything else for this version. Save your improvements for the final version.

Testing: You can test your program with the command `python Solver.py 3` or if you want graphics `python Solver.py 3 500`

Submission: Submit the two files MySolver.h and MySolver.cpp on the course Moodle page.

Final Version

Due: 6am Thursday, September 20. Late assignments will not be accepted.

Goal: Write an agent to solve the twenty-four puzzle that finds a solution (not necessarily optimal) as quickly as possible.

Testing: You can test your program with the command `python Solver.py 5` or if you want graphics `python Solver.py 5 500`

Submission: Performat a git commit and a git push to submit your files. Submitting your files will automatically test your code and post results on the contest page. This way you know how you perform relative to your classmates and the reference implementations. Note that the final grading will be using a different set of test instances. Programs will be given a maximum time of 30 seconds to initialize their algorithms and 30 seconds per puzzle.

Hints: To do well consider the following:

- Submit agents often. Test them, find improvements, document and submit again.
- Try basic algorithms/heuristics first.
- A few webpages worth looking at
  - `http://http://kociemba.org/fifteen/fifteensolver.html`
  - `https://heuristicswiki.wikispaces.com/N+-+Puzzle`
- I will keep adding ideas here, check back regularly.