Artificial Intelligence
Shikaku Puzzle Contest

Goal Write an agent to a Shikaku puzzle as quickly as possible. The optimal solution is the one that takes the minimum number of moves to solve.

Due Thursday, October 1 at 5pm.

Performance The performance of your agent on a series of example puzzles will be used to give you a performance grade on your project. Approximate performance grades will be posted on the contest website. The best performing agents will compete in the "class championship" for bonus opportunities.

Learning Outcomes There will be several learning outcomes that can be demonstrated in either your agent or your write-up.

Files All of the files can be found in your repository on git.cs.slu.edu and on the Canvas site.

Your module Save you module in the file `NAME.py` where name is your hopper username (all lower case). The you should name your solver class NAME which should be a child class of `ShikakuSolver`. You need to implement the function `solve`. See `BasicSolver.py` for an example.

Running the program Start Python and load all the modules:

```
from Shikaku import *
from ShikakuVisualizer import *
from Puzzles import *
from NAME import *
```

Only load the visualizer if you have installed Tkinter and cs1graphics and want graphics.

Create a visualizer If you plan on using the visualizer, `v = ShikakuVisualizer(900, True)` Where the 900 is the size of the screen you want, and the True indicates that you also want to see the options.

Create a solver `s = NAME(p1, 180, v)` Where p1 is the problem instance p1-p7 and 180 is the time limit in seconds. If you want to time how quick your algorithm is replace the v with None.

Run your solver `s.solution()`

See the video for advice on what to do if you don't have access to running graphics.

Write-up To meet the various learning outcome you need to provide a write-up describing the different versions of your program, how you tested them and what you did to improve your performance.

Submitting You should make sure your properly name solution and make sure it is submitted to the directory `contests/shikaku` in your class git repo. It will automatically be run by the software for the leader board for you to see how it performs against reference implementations and your classmates.

Testing Try your solver on different puzzles. If it solves it within the time limit it will return a Numpy array with the solution. If it unable to solve it it will return None.