

[To run]

```
# python solver.py puzzles/test1.txt
```

[Report]

- Inferences/Consistency

The nonogram puzzle is made consistent via various constraints:

1. Unary constraints on individual blocks (blocks are either rows or columns)
2. Binary (arc) constraints between all rows/columns.

Unary constraints are described by individual hints, and they reduce the domain range given to a smaller subset.

Binary constraints between rows/columns help ensure that propagation ties in with the validity of the puzzle. The only constraint that lies between 2 blocks are that the overlapping grid units must be equivalent.

- Propagation

Propagation happens via reducing worklists. Worklists are individual arc constraints that exist between rows/columns. As we reduce the worklist down, the rows/column domains are narrowed as well. And because the domains are affected, corresponding related blocks are also affected.

These new and changed domains which lead to a change in consistency in other blocks is essentially the engine that drives propagation of the puzzle.

- Algorithm

In this solver, we employ the AC-3 algorithm, otherwise known as the Arc Consistency Algorithm #3. It is deeply detailed and coupled with pseudocode at the following wikipedia article: http://en.wikipedia.org/wiki/AC-3_algorithm

Member(s) of the group

- Steven Goh Jian Wen (U087063E)