

Stable Matching Report

Sigurd Dinesen, Gustav Røder & Lars Yndal Sørensen

August 26, 2015

Results

The implementation can parse input files, and produce the expected output compared to the existing output files.

When the program isn't given any starting arguments, it will run all input files with this result, where flaws indicate differences between our output and the existing output files):

```
sm-bbt-in.txt: Flaws: 0
sm-friends-in.txt: Flaws: 0
sm-illiad-in.txt: Flaws: 0
sm-kt-p-4-in.txt: Flaws: 0
sm-kt-p-5-in.txt: Flaws: 0
sm-random-5-in.txt: Flaws: 0
sm-random-50-in.txt: Flaws: 0
sm-random-500-in.txt: Flaws: 0
sm-worst-5-in.txt: Flaws: 0
sm-worst-50-in.txt: Flaws: 0
sm-worst-500-in.txt: Flaws: 0
```

Implementation details

The implementation is based on the pseudo code from page 6 in Kleinberg and Tardos, *Algorithms Design*, Addison–Wesley 2005. The content of the input file is parsed into Person objects, which can be of either a Bro or Chick instance.

While there is a Bro instance which is unmatched, this Bro will propose to a Chick.

Even though the implementation is based on OOP, marriage proposals are evaluated by using two lookup tables – from spousal-preference order to Chick, and from Bro to Chick spousal-preference. This eliminates the need to traverse each Person's preference list, bringing the time complexity down to $O(n^2)$ by using constant-time lookups.