

Stable Matching Report

Michał Derdak, Tomas Vit, Kristinn þór Jónsson, Rafal Piotr Markiewicz, Alberto Martin Belloso, Rămniceanu Sabin and Albert Llebaria Holda

September 15, 2018

Results

Our implementation produces the expected results on all input-output file pairs. The current implementation has been developed with python.

Implementation details

The men's preferences are stored in a dictionary where each key consists of a man and its value consists on an ordered list of women.

¹ The same dictionary structure is used for the women's priority list. Each match is represented as a list where m, w are $[w, m]$ and such list is stored in the final matchings list. In order to improve the algorithm, once a man (let's call m to these handsome man) has been matched with a woman (w), w is removed from man's priority list. Therefore, if the match between m and w is unmatched due another man (m') has a higher priority than m in w , m will not ask again to w because w has been "discarded" for him.

The timing results from the our algorithm in the test case of 500 matches has an average time of 152.67 ms to find a free man who has not proposed to every woman. Also, at the same scenario, the implemented code takes an average time of 1012.81 ms to finish the algorithm.

With these data structures, our implementation runs in time $O(n^2)$ on inputs with n men and n women.

```
1 { "Wayne":  
    ["Chicago", "Boston", "Detroit"],  
    "Val": ["Boston", "Detroit", "Chicago"],  
    "Xavier": ["Detroit", "Boston", "Chicago"] }
```