Stable Matching Report

August, Johannes, Jonas, Martin, Michelle, Thomas
Group K

August 2020

1 Results

Our implementation produces the expected results on all input–output file pairs.

In example, on input `sm-bbt-in.txt`, we produce the following matching:
Sheldon – Amy
Rajesh – Priya
Howard – Bernadette
Leonard – Penny

2 Implementation details

All people are stored in list `persons` and the proposers (persons with even IDs) are at start added to list `free_m`. A person have an id, name, partner and field indicating if proposer.
The preferences are stored as a list for proposers and for the recipients they are stored as a dictionary mapping from proposers id to their rank.

While a proposer is free it is selected along with popping its preferred recipient yet to be asked. If the recipient doesn't have a partner the proposer and recipients are set as partners and the proposer is removed from the free list. If the recipient has a partner the preference is checked in constant time and a partner is changed if the recipient prefer the proposer to their current partner. If a new match is made the previous proposer is added to the free list and the new proposer is removed from it. When a proposer no longer has any recipients, which shouldn't happen, or if all proposers have a partner the partners are returned.
Each step in the iteration happens in constant time, under assumption that python's pop and append runs in constant time. The running time / number of iterations are in best case is $O(n)$ and the worst case is $O(n^2)$ on input with n proposers and n recipients.