Stable Matching Report

Group 14:
Cerda Toro, Oscar Felipe
Jakobsen, Andreas
Schade, Henrik Sloth
Schønemann, Adam Bjørn

Handed in at Monday 12, September 2016.

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

Implementation details
All men are stored in a queue, the men's preferences are stored in a stack by preference.
The women are stored in an ArrayBuffer, and their preferences are stored in an array field. To lookup the preference a woman has for a man, his ID is divided by 2 and in the array a number indicating the woman's preference is stored at this index. That means a woman can compare two men by looking up her preferences towards them in constant time.
For example, if a man with ID 3 is proposing to a woman, and he is her most preferred man, the second element in her preference array (because 3/2 = 1) will result in the number n. Thus, the array constitutes a mapping from (floor(id/2) -> p) where p <= n and a larger p indicates a stronger preference.

A man is taken from the queue until no more men are left in the queue. Once a man is taken from the queue, we propose to the first woman on his preference list. The woman is removed from his preference list. If she is not engaged or the preference of the man proposing has a higher preference, she accepts and we will update their engaged status with their respective id. A rejected man is re-queued to the list of men.
Since every man can at most ask every woman, the upper bound of the running time is n^2. Because we index the preference arrays, the look-ups are done in constant time, which again makes the total running time $O(n^2)$ with n men and n women.