Mitchell Dennen
Shahin Safarov
Sowmya Kodityala

Assignment 1: Stable Marriage:

1. From the moment that g receives her first proposal, g remains engaged. Also, her sequence of partners gets better and better (in terms of her list of preferences).

Once a g gets engaged(line 7), g will either remain with the same partner b, or get a new partner b*(line 10) only if g prefers b* to b(line10)

2. The sequence of g's to whom a particular b proposes gets worst and worst (again, in terms of his list of preferences).

If a b* prefers a g that is engaged, and the g prefers b* to her b (line 9), the new b* will have to engage a g lower down his rank (line 6).

3. The following is an invariant of the G-S Algorithm: if b is free (not engaged) at some point in the execution of the algorithm, then there is a g to whom he has not yet proposed.

As we loop through all the b's (line 2) there is always exactly 1 g that gets engaged (line 7,10), this will ensure that there is always the same number of b's and g's left un paired by the end of the first for loop.

4. The set of pairs M at the end of the execution of the algorithm constitutes a perfect matching. (Start by defining what a "perfect matching" is.)

Perfect matching is when every b and g is monogamous. As we loop through all the b's (line 2), b will either pair with an unmatched g (line 7), or g will leave paired b for b* (line 10). In this case the unmatched b will become b* (line 11) and match with the next preferred available g (line 12). By the end of each iteration of the first for loop, every iterated b is engaged to exactly one g.

5. The set of pairs M at the end of the execution of the algorithm constitutes a stable matching. (Start by defining what a "stable matching" is.)

A stable matching occurs when every pair has a perfect matching and there is no blocking pairs. A blocking pair is when a b and a g are not pairs but they both prefer each other.
In each iteration of the loop, each b gets his preferred g as long as g is not engaged (line 7). If g is engaged, and g prefers b* to current b (line 9) then b* engages g (line 10). This ensure b and g always end up with their best available preferred matching.

6. **Give an example of a B,G with corresponding lists of preferences for which there is more than one stable matching.**

b1: G:4,3,1,4	g1: B: 2,3,1,4
b2: G: 2,3,1,4	g2: B: 2 4 3 1
b3: G: 4,2,1,3	g3: B: 1 4 2 3
b4: G: 2,3,1,4	g4: B: 4 1 3 2

M= {(b1,g4), (b2, g2), (b3, g1), (b4, g3)}
M = {(b1,g3), (b2,g2), (b3,g1), (b4,g4)}

7. Recall the definition of a feasible pair in the textbook (pg. 17). Let's say that g is the best feasible pair for b, if (b, g) is a feasible pair, and there is no g′ such that:
g′ <b g and (b, g′) is also a feasible pair.
For any given b, let B(b) be b's best feasible pair. Finally, let M∗ = {(b, B(b)) : b ∈ B}.
Show that the G-S Algorithm yields M∗.

In fig 8.1 below, M3 = {(b1,g2),(b2,g1)(b3,g3). This holds true to M* = {(b, B(b) : b ∈ B)} because b1 is engaged to his number 1 g, b2 is engaged to his number 1 g, and b3 tried to engage his number 1, but she was already engaged to her number one, so he tried to engage his number 2, but she was engaged to her number one, so he engaged his number 3 because she was not engaged. Thus we achieved a boy optimal feasible pair for each b in B.

8. Show that any re-ordering of B still yieldsM∗, that is, the G-S Algorithm is independent of the order of the boys.

(Mitchell) -

B1: G:2,3,1	G1: B:2,3,1
B2: G:1,3,2	G2: B:1,2,3
B3: G:2,1,3	G3: B:3,2,1

Order: b1,b2,b3	Order: b3,b2,b1
1: M1 = {(b1,g2)} 2: M=M1, b* = b2, M2 = {(b1,g2),(b2,g1) 3: M=M2, b* = b3, M3 = {(b1,g2),(b2,g1)(b3,g3) fig.8.1	1: M = {(b3, g2)} 2: M = M1, b* = b2, M2 = {(b3,g2)(b2,g3)} 3: M=M2, b* = b1, M = {(b2,g3)(b1,g2), b* = b3, M3 = {(b3,g3)(b2,g1)(b1,g2)

Order of b's will not matter because although b* will always check his preferences in his rank order, g must prefer him to her current b.

9. Show that in M∗, each g is paired with her worst feasible partner.

Because we iterate through the b's, and b always gets preference to a g, as long as g prefers the b to which it currently is engaged to, g will never get ranked preference unless g is lucky.

10. Assess the running time (complexity) of the algorithm in terms of Big-Oh complexity.

O(n^2) because through each iteration of b's, it can go through each g to find a suitable match.

Algorithm 1.7 Gale-Shapley

1: Stage 1: b_1 chooses his top g and $M_1 \longleftarrow \{(b_1, g)\}$
2: **for** $s = 1, \ldots, s = |B| - 1$, Stage $s + 1$: **do**
3:     $M \longleftarrow M_s$
4:     $b^* \longleftarrow b_{s+1}$
5:     **for** b^* proposes to all g's in order of preference: **do**
6:         **if** g was not engaged: **then**
7:             $M_{s+1} \longleftarrow M \cup \{(b^*, g)\}$
8:             end current stage
9:         **else if** g was engaged to b but g prefers b^*: **then**
10:             $M \longleftarrow (M - \{(b, g)\}) \cup \{(b^*, g)\}$
11:             $b^* \longleftarrow b$
12:             repeat from line 5
13:         **end if**
14:     **end for**
15: **end for**