

Proof of 1(a). Trivial. Suppose a man m and a woman w are single. Then the pair (m, w) violates the definition of a stable matching. \square

Proof of 1(b). Since men will eventually run out of proposals, the algorithm fails to terminate if and only if there exists both a single man and a single woman by the time all men have proposed to all women. Suppose for contradiction that this is the case; call the man m and woman w . Just like in lecture, we note that once a woman has been proposed, she will never become single again.

- If there are more men than women, then w never received a proposal; in particular m never proposed to w . Therefore m can still propose to w , reducing the number of single men by 1.
- If we start with equal numbers of men and women, the proof is identical (as shown in lecture).
- If there are fewer men than women, and if m remains single after proposing to all women, the only possibility is if all women rejected him. However, this is impossible since each man can only pair with one woman at one time, and there will always be ≥ 1 single women, who would not reject m . \square

Proof of 1(c). We represent men by $\{1, 2, 3\}$ and the woman by $\{A, B\}$. We assume all men prefer A over B and all women prefer 3 over 1 over 2. We run the algorithm as follows.

- Pick a single man: 1.
 - 1 proposes to A .
 - A is single — accept.
 - Current matching: $\{(1, A)\}$.
- Pick a single man: 2.
 - 2 proposes to A .
 - A rejects 2.
- Pick a single man: 2 again.
 - 2 proposes to B .
 - B is single — accept.
 - Current matching: $\{(1, A), (2, B)\}$.
- There is no single woman anymore. Terminate program.

Note that 3 is single and A prefers 3 over her current partner 1. That is, the pair $(3, A)$ renders the matching $\{(1, A), (2, B)\}$ unstable. \square

Proof of 1(d). We have proven the case in class when there are as many women as men, so we now suppose there are strictly more women. Suppose the output is not stable. By definition, there exist a man m and woman w such that

(*) m is single OR m prefers w over his partner w' , and

(**) w is single OR w prefers m over her partner m' .

If m is single, then w must not be by 1(b). Therefore, w prefers m over her partner m' . However, this violates the fact that each woman will only get better partners (according to their rankings) every time they dump someone, so it is impossible for w to end up with someone she considers worse than w .

If m is not single, then m must prefer w over w' . That he did not match with w implies either w rejected or dumped m . Suppose w rejected m . This means at that moment of rejection, w was not single and had a partner better than m , contradicting (**). Suppose alternatively that w dumped m . By the same reasoning, w was neither single nor preferred m over her new partner who, according to the definition, is at most as good as m' , her current partner. Contradiction again.

Therefore the output matching must be stable. □

Proof of 2(a). Let $\{1, 2\}$ be men and $\{A, B\}$ women, with 1 preferring A , 2 preferring A , A preferring 2, and B preferring 1. Then $\{(1, B), (2, A)\}$ is a stable matching but neither men-stable nor women-stable. □

Solution for 2(b). The algorithm goes as follows:

```

1 Start with empty assignment
2 for each student s:
3     assign s their top available choice r
4     mark r as occupied
5 end

```

It is easy to see that this results in a student-stable matching. Suppose not, that students a, b prefer each other's rooms and WLOG assume student a picked theirs first. Since student b picked their room later, b 's room must be available at the moment a picked their room, meaning that a did not pick their top priority, contradiction.

For complexity, in the worst-case scenario, student j will find their top $j - 1$ choices occupied by the previous $j - 1$ students, and so the total runtime is $\mathcal{O}(n(n - 1)/2) = \mathcal{O}(n^2)$.