1. Define the problem.
2. Define the algorithm.
3. Bound the worst-case run-time.
4. Prove correctness.

Definition of Stable Marriage. A instance of STABLE MARRIAGE is specified by a positive integer n that implicitly defines a set of boys $b_1, b_2, \ldots, b_n$ and a set of girls $g_1, g_2, \ldots, g_n$, along with a collection of *preference lists*: for each boy, a total ordering of the girls, and for each girl, a total ordering of the boys.

A *matching* M is a collection of boy/girl pairs so that each boy and each girl is in exactly one of the pairs. For each pair (b, g) in M, we say M *matches* boy b and girl g. We say a pair (b, g) is *unstable* if b and g are *not* matched, and both b and g prefer each other to their matched partner in M. The matching M is *stable* if there are no unstable pairs. The output for the problem is a stable matching, if there is one.

The "boys-propose" algorithm

1. Maintain a partial matching, initially empty.
2. Repeat until each boy is either matched or has been rejected by all girls:

3.1. Choose any unmatched boy b that has not been rejected by all girls.
3.2. Boy b *proposes* to the girl g that he prefers most, among all girls that have not yet rejected him.
3.3. If girl g is unmatched, she accepts the proposal, and b and g are matched.
3.4. Else, if g prefers her current match b' to b, she rejects b, and stays matched to b'.
3.5. Else, g accepts: she rejects b', and becomes matched to b instead of b'.
4. Finally, return the (possibly incomplete) matching formed by the currently matched pairs.

Lemma 1 (termination)**.** *In any execution of the algorithm, there are at most n^2 iterations.*

Proof (long form).

1. Consider an arbitrary execution of the algorithm. *to show* $\forall$
2. In each iteration, one boy proposes to one girl.
3. By inspection of the algorithm, no boy proposes to the same girl twice.
4. By the previous two lines, there are at most n^2 iterations (one for each boy/girl pair). $\square$

Since the algorithm can be implemented so that each iteration takes constant time, the lemma implies that the algorithm can be implemented to run in $O(n^2)$ time. Next we show the algorithm is correct.

Lemma 2 (utility lemma)**.** *After any execution of the algorithm, every boy and girl are matched.*

Proof (long form).

1. Consider an arbitrary execution of the algorithm. *to show* $\forall$
2. By the previous lemma, it terminates.

3.1. Assume for contradiction that it leaves any boy and/or any girl unmatched. *proof by contradiction*
3.2. Let b and g be any unmatched boy and girl. (They exist, since there are n boys and n girls.)
3.3. By the termination condition of the algorithm, boy b was rejected by g at some point.
3.4. By inspection of the algorithm, when b was rejected by g, girl g was matched.
3.5. Likewise, g remained matched for the remainder of the execution.
3.6. This is a contradiction, since g is unmatched at the end.
4. By Block 3, all boys and girls are matched at termination. $\square$

Lemma 2 shows that the algorithm returns some matching. We need to show that the matching is stable.

Lemma 3 (correctness)**.** *Any execution of the algorithm returns a stable matching.*

Proof 2 (long form).

1. Consider an arbitrary execution of the algorithm. *to show* $\forall$
2. Let M be the matching it returns (by Lemma 2).

3.1. Assume for contradiction that M is not stable. *proof by contradiction*
3.2. Let (b, g) be an unstable pair. (The pair exists, by definition of stable matching.)
3.3. Let g' and b' be the partners of b and g, respectively, in M.
3.4. By definition of unstable pair, b prefers g to g', and g prefers b to b'.
3.5. Boy b proposed to g' in some iteration (because he is matched to g').
3.6. He must have been rejected by g before that iteration (or he would have proposed to g instead).
3.7. In the iteration that g rejected b, she must have been matched to a boy that she prefers to b.
3.8. In any subsequent iteration, if she changed to a new partner, she prefers the new partner even more.
3.9. So, at the end, she must be matched to a boy that she prefers to b.
3.10. So the pair (b, g) is not unstable. This is a contradiction (with line 3.2).

4. By Block 3, M is stable. $\square$

See also:

- KT (Kleinberg & Tardos) Chapter 1.1 (text)
- https://en.wikipedia.org/wiki/Stable_marriage_problem — wikipedia
- https://archive.org/details/ucberkeley_webcast_8qhxIAWp-zk — video lecture by Umesh Vazirani at Berkeley.
- Followup lecture notes on boy-optimality of the boys-propose algorithm.