CS 180

Max Smiley

Spring 2021

This is an introductory course to algorithms that I took at UCLA. After the first few introductory lessons, we cover graphs, greedy algorithms, divide and conquer, dynamic programming, NP completeness, and randomized algorithms. In each section, I'll go over definitions, theorems, and proofs. This is a relatively informal set of notes, mainly intended to be referenced by myself in the future.

The Stable Matching Problem

The *Stable Matching Problem* is an idealism of many real world problems, in which we want to find a reasonable pairing between members of two different groups. For example, algorithms developed to study this problem are implemented in medical school, where students get residencies and are matched to various institutions.

The Stable Matching Problem is presented as follows. Suppose we have two groups of equal size, for example, we take these groups to be men and women, where $M = \{m_1, \dots m_n\}$ and $W = \{w_1, \dots w_n\}$. Each $m \in M$ and $w \in W$ also have a preference list of the opposite sex. The goal is to create a one to one pairing between members of the groups, such that the matching is stable.

Def: a *perfect matching* between n men M and n women W is a bijection $S = \{(m_1, w_i), \dots (m_n, w_j)\}$ between elements of M and W. That is, for each $m_i \in M$, there exists a unique $w_j \in W$ such that $(m_i, w_j) \in S$, and vice versa.

Def: a *rogue pair* is a stable matching S such that $\exists (m_i, w_j), (m'_i, w'_j) \in S$ such that m_i prefers w'_j to w_j, and w'_j prefers m_i to m'_i.

Def: a *stable matching* is a perfect matching such that there are no rogue pairs.

Gale-Shapley Algorithm:

```
while $\exists$ unmatched $m \in M$ :
    select $m \in M$, find highest ranking woman $w \in W$ not yet proposed to
    if $w$ is unmatched
        add $(m, w)$ to $S$
    else if $w$ prefers her current man $m'$ to $m$
        reject $m$
    else if $w$ prefers $m$ to her current man $m'$
        remove $(m', w)$ from $S$, add $(m, w)$ to $S$
```

Lemma A: Once w is matched, she will never be unmatched.
Proof: if w is unmatched, then she will either reject m, or break her relationship with m' in favor of m. In either case, w will still be matched.

Lemma B: Once matched, $m's$ partners will only decrease in favorability.
Proof: if m has previously had partner w, then m had proposed to w, and thereafter may only propose to women lower on his priority list. Thus, he can only match with those women.

Lemma C: Once matched, $w's$ partners will only increase in favorability.
Proof: suppose w is matched with m. Then, she will only accept a new man m' if she prefers m' to m.

Claim: The GS algorithm terminates.
Proof: the algorithm iterates over n men, each of which having n women to potentially propose. Thus, the algorithm terminates after at most n^2 steps.

Claim: The GS algorithm outputs a perfect matching.
Proof: since the algorithm assigns 1 to 1 pairs, if there is not a perfect matching by the end of the algorithm, then $\exists m \in M$ such that no $(m, w) \in S$ for $w \in W$. However, the algorithm wouldn't have ended if there were still an m leftover at the end, it would have assigned him to w.

Claim: The GS algorithm outputs a stable matching.
Proof: Assume, by way of contradiction, that the GS algorithm did not output a stable matching. Then $\exists (m, w), (m', w') \in S$ such that m prefers w' to w and w' prefers m to m'. If this were the case, then m would have proposed to w' before w. w' would have either rejected m (if she had a better mate) or accepted him, but in either case, by Lemma C, w' would never have matched with m' thereafter, as she prefers m to m'.

Claim: *Proof:*