ASSIGNMENT - 01

COMP 544 - ALGORITHM

ALGORITHM ASSIGNMENT GROUP - 03

Please submit one assignment per group; form the groups at the beginning of the course, and work together on all assignments.

Prove the following claims about the Gale & Shapley Algorithm (Algorithm1.7). This sequence of claims builds toward a proof of correctness of the G-S Algorithm.

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).**

   Ans. According to his list of preferences, bs+1 makes romantic proposals to the females. If one of the girls accepts, and she was free, she becomes the new girl with a partner. If not, some b.

   We keep making the same argument because b1,..., and bs stopped paying attention. The g's disengage only if a better b offers, so pMs+1 (gj) <j pMs(gj).

2. **The sequence of g's to whom a particular b proposes gets worst and worst (again, in terms of his list of preferences).**
   Ans. When bs+1 asks girl gj to marry him, gj accepts if either gj is not now engaged or is already engaged to a boy b who is less desirable, or bs+1 j b. In the case where gj prefers bs+1 over her current partner b, then gj breaks off the engagement with b and b then has to search for a new partner. Whether or not b's initial proposal to g was accepted, the partners of g could only have been better after that. Therefore, b doesn't need to attempt again. As a result, the list of gs that a specific b proposes gets worse and worse.

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.
   Ans. Contrarily, suppose that there is a male bi who is single and has made engagement proposals to every girl on his list at some point during the process. This implies that every g on his list was already engaged to a more desirable b since every g on his list refused bi. Since there are no free girls and we have n girls engaged to n 1 boys, this deviates from our initial hypothesis.

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.)**

Ans. A perfect match occurs when there is a 1:1 match between the boys and girls, and each boy is specially engaged to a different girl. Assume that at the end of the algorithm's execution, this is not the case. A one-to-many relationship will never occur since the algorithm prevents a g from engaging a b without first being free (and disengaging if necessary). This indicates that at the end of the algorithm's execution, a boy named bi and a girl named gj are both free. If gj is single, it signifies that she has never been proposed to; nevertheless, since bi would have done so for every g on his list, he must have done so for gj. This results in a contradiction, hence a perfect match must be true.

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.)**

   Ans. If a matching M doesn't have any blocking pairs, it is stable.

   Imagine that we have the blocking pair ((b, g')) b, g.),(b', g) ⊆ M_n, but g prefers b to b' and b to g, respectively). Either b comes before or after b'. If b came first, g would have been with b or someone better when b' arrived, therefore g would not have existed. having tied the knot with b'. However, because (b ', g) is a pair, b could not have come after b'because g hasn't received a greater offer than the one provided by b '. The matching is stable because there is no blocking pair (b, g) in either scenario and we instead receive an impossibility.

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

   Ans. Take B = {b1, b2} such that g2 <1 g1 and g1 <2 g2 and G = {g1, g2} such that b1 <1 b2 and b2 <2 b1.

   Boy optimal M = {(b1, g2),(b2, g1)}

   Girl optimal M0 = {(b1, g1),(b2, g2)}

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′ <bg 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∗.

   Ans. Contrarily, suppose that B is the first boy who receives a rejection from B, his best possible partner (b). This indicates that B(b) already has some pairs with b' and that B(b) prefers b' to b. B(b) is also at least as appealing to b' as his own best possible companion (since the proposal of b is the first time during the run of the algorithm that a boy is rejected by his best feasible partner). We know (by definition) that there exists some stable matching S where (b, B(b)) is a pair because B(b) is the best viable mate for b. However, since B(b) is already taken by b, whoever b'is matched with in S, let's say g ', b'prefers B(b) to g '. The best viable partner of b'is ranked (by b' of course) at most as high as B(b). Because b ', B(b) prefer each other to the partners they have in S, this results in a blocking pair. As a result, each b B must be coupled with B(b), which is the pair's best practical mate.

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

Ans. It is clear from question number 7 that the boys' order is irrelevant because there is a specific boy-optimal pairing (each boy b gets matched with B(b)), which is independent of the boys' order.

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

Ans. We exploit the fact that each boy is paired with his best viable mate to demonstrate that the Gale-Shapley algorithm pairs each girl with her worst feasible companion (which we just showed in problem 7). Once more, we argue using contradictions. If g is paired with b in a stable matching S, and g prefers b' to b, then (b', g) is the output of the Gale-Shapely algorithm. We know that in S we have (b', g'), where g' is not higher on the preference list of b' than g, and since g is already matched with b, we know that g' is really lower. This is because each boy is paired with their best practicable partner. This indicates that S is unsteady because b and g form a blocking duo because they prefer each other to their partners.

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

Ans. The computation takes O(n^3) time. This is due to the fact that at any stage s+ 1, there is an upper bound of (s + 1)^2 steps, giving each stage an O(n^2) time complexity (this is the situation when a g is already engaged and pushes another b to look for a new partner). Additionally, because there are n phases in total, the total time complexity is O(n^3).