Assignment-1

Name: DYUTI MANGAL

Roll No: 190101035

Q1. What is a greedy algorithm?

An algorithm is greedy if it builds up a solution in small steps, making a decision at each step to obtain the most obvious and immediate benefit. This means that it makes a locally-optimal choice in the hope that this choice will lead to a globally-optimal solution. The Greedy algorithm has only one oppurtunity to compute the optimal solution so that it never goes back and reverses the decision.

Q2. Does the greedy technique always work?

Greedy calculations are very effective in certain problems and generally quite easy to come up with. Notwithstanding, in numerous problems, a greedy strategy does not produce the correct answer. For greedy algorithms you have to work much harder to understand correctness issues. Even with the correct algorithm, it is hard to prove why it is correct.

Q3. Define bipartite graph.

A bipartite graph, is another name for a "two-colourable graph", when we can assign two colours to vertices and make sure that no adjacent vertices have the same color. More formally, it is a set of graph vertices disintegrated into two disjoint sets in such a manner that two adjacent graph vertices don't happen to occur in the same set.

Q4. What is perfect matching in a bipartite graph?

Let us say G is a bipartite graph and S is a set of edges in G in such a manner that no two edges in G have a vertex in common. Such an S is called a matching in G. If there's no vertex in G, that is not covered by some edge in S, then such a matching S is called a perfect matching.

Q5. Define the stable matching problem in a bipartite graph.

The stable matching problem is defined as, given a list of preferences for each element, our aim is to devise a matching between two sets of elements, which is 'stable', meaning there should not exist edges ab, $cd \in M$ (where $\{a, c\} \in A$ and $\{b, d\} \in B$) such that d is preferred over b by a, and a is preferred over c by d.

Q6. What is the difference between a perfect matching and a stable matching?

Let us consider two sets A and B of the same size. A 'perfect matching' between the two sets would be one such that every element of A is connected to precisely a single element of B and vice versa. On the other hand, 'stable matching' means we have a list of every element's preferences from the opposite set, and that in the matching created, no two elements of opposite sets prefer each other over their currently allotted partner.

Q7. What is a blocking pair in stable matching problem?

Let us say A and B are two sets of vertices with the matching M. A 'blocking pair' is a pair of elements (a, d) if there exist $ab, cd \in M$ (where $\{a, c\} \in A$ and $\{b, d\} \in B$) such that a prefers d over b, and d prefers a over c. The existence of a blocking pair in a matching introduces instability.

Q8. In the stable matching problem with complete preference lists, does there always exist a stable matching?

Given two disjoint set of vertices of the same size, and the preference list of all elements in those sets, we can always obtain a stable matching utilizing the Gale-Shapley Algorithm.

Q9. Write pseudo-code of Gale-Shapley algorithm for stable matching problem.

Let M be the set of men, and one individual man be represented by m. Let W be the set of women, and one individual woman be represented by w.

At first, all $m \in M$ and all $w \in W$ are free.
While (there exists a man m who is free and hasn't proposed to every woman)
    Choose such a man m_1
    Let w_1 be the highest-ranked woman in m's preference list to whom m hasn't proposed yet.
    If w is free then
        (m, w) become engaged
    Else w is currently engaged to m'
        If w prefers m' to m then
            m remains free
        Else w prefers m to m' then
            (m, w) become engaged
            m' becomes free
        Endif
    Endif
Endwhile
Return S, the set of engaged pairs

Q10. Name five applications of Gale-Shapley stable matching algorithm.

Some examples of applications of G-S Algorithm:
1) Finding an appropriate opponent in a one on one online game.
2) Assignment of large number of seats of colleges/schools to students.
3) Pairing transplant donors with patients.
4) Online dating platform service.
5) Making suitable pairs for a class project.

Q11. Let the preference list of 3 men and 3 women be as given. What can be a stable matching here?

The stable matching obtained through G-S Algorithm is: (M_1, W_1) , (M_2, W_3) , (M_3, W_2)

Q12. Given Students preference order, and Teachers Preference. Explain how the GS algorithm would work on this example. Also give the final stable pairs.

- Assuming the students do the proposing, S_1 asks to be with T_1, and since T_1 is free, they are paired.
- Now S_2 asks to be with T_1. So T_1 being already paired has to check their preference list, according to which, their current match S_1 ranks higher than S_2, so S_2 remains free.
- Now S_1 is already paired and S_2 starts the proposing again. S_2 asks to be with the next person in her list that she hasn't already been rejected by, T_2. Since T_2 is free, the pairing is made.
- Now all the pairings are complete.
- Final matching : (S_1, T_1) , (S_2, T_2)

In this particular problem, whether we start proposing from Students' side or from Teachers' side, we obtain the same stable matching.

Q13.Is the Gale-Shapley algorithm for the Stable Matching problem a greedy algorithm? Why or why not?

In the working of the Gale Shapley algorithm, in each step the proposers propose to their most preferable partner whom they've not been rejected by already, that is, they're making a locally optimum choice in hopes of achieving a globally optimum solution. This exactly is the nature of a Greedy algorithm. Hence, the Gale-Shapley algorithm for Stable Matching Problem is a greedy solution.

Q14. How can you implement G-S algorithm efficiently? Explain.

Let us assume men are labelled 1,2...,n and women as 1',2',...n'.

- We make a doubly linked list to store the free men, since this list only needs to be accessed in any particular direction, not random elements.
- Two more arrays are created to store the engagements. 'women[m]' and 'men[w]' with all entries set to 0 initially indicating not matched. If m is paired with w, then women[m]=w and men[w]=m.
- Next for each man, we maintain a list of their preference on women in order.
- And one list called 'count' to store the number of proposals made by each man.
- Last for each woman we create an inverse preference list, meaning the indices represent man number and the value represents his preference number.
- As a result of all this, it takes $O(n)$ time of preprocessing and the women's preference is accessed in constant time. Total n such iterations makes it $O(n^2)$.

Q15. What are the data structures used in the implementation of Gale-Shapley Algorithm?

We can use **Doubly Linked Lists** or **Queues** to store the information about free men since they are both dynamically changing data structures and we only need to access this detail in a particular direction and not random elements. We use **Arrays** to store our preference orders making it possible to access a certain preference in constant time.

Q16. Does G-S algorithm guarantee to find a stable matching ? If yes, give proof of correctness.

To prove this, let's proceed with assuming the contrary that G-S Algo produces an unstable matching.

This implies there exist two elements a and b' not matched by the algorithm, such that a prefers b' to its current partner b and b' prefers a to its current partner a'.

According to this prefernce described, a must have proposed to b' before b since b' ranks higher than b on the preference list of a.

But any offer is only rejected if the one receiving the offer already is paired to someone of higher preference. This means b' should've been paired with someone more preferable than a if b' rejected a.

This implies b' prefers a' over a, which is a **contradiction**.
Hence we can say that, yes **Gale-Shapley algorithm always produces a stable matching.**

Q17. What is the time complexity of Gale - Shapley Algorithm ? Explain the analysis.

Let's say all men are standing in a queue in the order $m_1, m_2, ..., m_n$ initially. The algorithm starts with the front of the queue, and those men who become free during the execution of the algorithm join at the end of the queue.

To calculate worst case time complexity, assuming w_1 is the first preference of all men, w_2 the second and so on. And for women, the first preference is m_n, second m_{n-1} and so on.
So now, m_1 proposes to w_1 and is accepted. Then m_2 proposes to w_1 and is accepted leaving m_1 free. This could go on until m_n.
At the end of this w_1 and m_n are engaged and all others are free, and a total of n proposals were made.

Continuing in this manner, the number of proposals to be made in next iteration are $n-1$. And $n-2$ for the next.

Hence the number of total proposals made by the algorithm are :

$$n + (n-1) + (n-2).... + n = \frac{n(n-1)}{2}$$

Giving us a time complexity of $O(n^2)$.

Q18. Given an instance of stable matching problem, do all executions of G-S algorithm yield the same stable matching? Explain your answer with proof.

Yes all executions of G-S algortihm result in the same stable matching. To prove this, let us proceed with contradiction, that some execution λ produces a matching S in which some man is not paired with his best valid partner. Since men propose in decreasing order of preference, let us consider the first time in the execution λ that a man m is rejected by a valid partner w, this means, w is the one at the top of the preference list of m.
This also means in this case, w is currently paired with some m' whom she prefers over m.

Now as w is a valid partner of m, there exists a stable matching S' containing the pair (m, w). And let m' be paired with some w' in this matching.

Now we know that in execution λ, the rejection of m by w was the first rejection, meaning m' had not been rejected by any valid partner at that point when he became engaged to w. We know that w' is also a valid partner for m' as in S', it means m' prefers w over w'.

We also know that w rejected m in favour of m' and prefers m' over m. But $(m', w) \notin S'$, which implies S' is an unstable matching.

Hence our initial assumption that S' is a stable matching is contradicted. Hence it is proved that every execution of the G-S Algorithm produces the same stable matching.

Q19. In the Gale-Shapley algorithm, with n men and n women, what is the maximum number of times any woman can be proposed to? Justify.

Let's say w_1 is the top preference of all men in our set, and the preference of w_1 is in the order $m_n > m_{n-1} > .... > m_2 > m_1$.

First m_1 proposes to w_1 and is accepted. Then m_2 proposes to w_1 and is accepted leaving m_1 free. This process continues until m_n proposes to w_1 and is accepted and no one further proposes to w_1.

Hence w_1 received n total proposals. **Maximum number of propsals a woman can receive is n.**

Q20. Define best valid partner and worst valid partner with respect to stable matching problem.

For any woman w, m is said to be her valid partner if there exists a stable matching containing the pair (m, w).
The worst valid partner for w is m, if he is a valid partner for w but no one ranking lower than m on her preference list is a valid partner. The best valid partner for m is w, if she is a valid partner for m but no one ranking higher than w on his preference list is a valid partner.

Q21. (a) In the execution of Gale-Shapley algorithm which follows the policy of man proposing to woman, a man's partner becomes worse and worse whereas a woman's partner becomes better and better. Explain.

During the execution of the algorithm, from a woman's view, she is free until no one has proposed to her. When any man m proposes to her, she is engaged. Now this engagement breaks only if she receives a proposal from a man more preferable to her than m is, meaning her partner keeps getting better and better.

From a man's view, he first of all proposes to a woman high on his preference list. If he gets accepted right now, he may still be rejected later if this woman receives a better proposal, in which case he would have to move on to the next woman on his list and so on. This shows that his partner keeps getting worse and worse as the algorithm progresses.

(b) Consider a stable matching S* generated by G-S algorithm for an instance of the stable matching problem. Show that each woman is paired with her worst valid partner.

Let's say there is a pair (m, w) in $S*$ where m is not the worst valid partner for w. This means there is a stable matching S' containing a pairing of w with m' whom she prefers less than m. Now in S', m is paired with some w'. But since w is the best valid partner of m, and w' is a valid partner of m, we arrive at the conclusion that m prefers w over w'. This would mean that (m, w) is a blocking pair in S', contradicting the initial assumption that S' is a stable matching. Hence each woman is for sure paired with her worst valid partner.

Q22. Write an algorithm where each woman is paired with her best valid partner.

Since it is known that in the generic G-S Algorithm execution for the Stable Marriage Problem, all men get their best valid partner and all women get their worst valid partner when men do the proposing, we can achieve an algorithm where women receive their best valid partner by having women do the proposing.

Let M be the set of men, and one individual man be represented by m. Let W be the set of women, and one individual woman be represented by w.

At first, all $m \in M$ and all $w \in W$ are free.
While (there exists a woman w who is free and hasn't proposed to every man)
    Choose such a man w_1
    Let m_1 be the highest-ranked woman in w's preference list to whom w hasn't proposed yet.
    If m is free then
        (m, w) become engaged
    Else m is currently engaged to w'
        If m prefers w' to w then
            w remains free
        Else m prefers w to w' then
            (m, w) become engaged
            w' becomes free
        Endif
    Endif
Endwhile
Return S, the set of engaged pairs.

Q23. (a) Suppose we are given an instance of the stable matching problem for which there is a man m who is the first choice of all women. Prove or give a counterexample: In any stable matching, m must be paired with his first choice.

Assuming the contrary, let's say man m is matched with some w' who is not on the top of his preference list. Since men propose in reducing order of preference, he would have proposed to his first preference w first. If he isn't currently matched with w it means he was rejected by her. But w would only reject m if she was already paired with someone she likes more than him. Here, we arrive at a contradiction since m was supposed to be the top preference of all women. Hence Proved that m must be paired with his first choice.

(b) If all the men have the same preference list of women, then briefly discuss about the final matches. You can explain with an example.

Assuming every man's preference list to be $\{w_1 > w_2 > w_3 > ... > w_n\}$. So clearly, all men will propose to w_1 and hence she will get matched with her top preference, say m_1. Now every free man will propose to w_2, and so she would get her top preference if its not m_1 and her second best preference if her top preference is m_1. Apparently w_2 is matched with the top man in her preference list that is free, her best valid partner. Thus we keep going down in this manner and all women get the best person on their list who is free, they get their best valid partner.

(c) Assume that the preference list of all women are same and so is the case for men. Comment on the pairs formed after applying GS algorithm.

Let's say that the preference list of every man is $\{w_1 > w_2 > w_3 > ... > w_n\}$ and similarly the preference list of every woman is $\{m_1 > m_2 > m_3 > ... > m_n\}$. So, m_1 proposes first and gets matched with w_1 her top preference and she would reject all the other men because m_1 is her top preference. Now, m_2 proposes w_1 and is rejected, so he proposes to w_2 and they get matched. Similarly m_3 and w_3 would be matched and so on. Thus every man m_i is matched with w_i. Basically the common preference lists are horizontally matched.

Q24. Suppose S is a stable matching for a given instance I of the Stable Matching problem, not necessarily the one produced by the Gale-Shapley algorithm.

(a) Does the matching S necessarily become unstable if we reverse the preference list of all the women (and keep the men's preferences intact)?

No, the matching S does not necessarily become unstable on reversing the preference list of women. See a counter-example:
$M_1 : \{W_2, W_1\}$, $M_2 : \{W_1, W_2\}$, $W_1 : \{M_1, M_2\}$, $W_2 : \{M_1, M_2\}$

We get the pairs $\{M_1, W_2\}$ and $\{M_2, W_1\}$ using the G-S Algorithm.
After reversing the preferences of women, new preference lists are:
$M_1 : \{W_2, W_1\}$, $M_2 : \{W_1, W_2\}$, $W_1 : \{M_2, M_1\}$, $W_2 : \{M_2, M_1\}$
We still get the same pairs on applying the G-S Algorithm. No blocking pair is introduced.

(b) What happens when we reverse the preference lists for all the men as well as all the women - does S necessarily become unstable? Justify your answer in each case.

No, the matching S does not necessarily become unstable on reversing the preference lists of both men and women. See a counter-example:
$M_1 : \{W_1, W_2\}$, $M_2 : \{W_1, W_2\}$, $W_1 : \{M_1, M_2\}$, $W_2 : \{M_1, M_2\}$
The following pairs are achieved by G-S Algorithm $\{M_1, W_2\}$ and $\{M_2, W_1\}$.
On reversing the preferences of both men and women:
$M_1 : \{W_2, W_1\}$, $M_2 : \{W_2, W_1\}$, $W_1 : \{M_2, M_1\}$, $W_2 : \{M_2, M_1\}$
Applying GS Algorithm gives the same results and no blocking pairs are formed.

Q25. In an execution of the standard GS algorithm, the partners to which a woman gets engaged gets better and better as the algorithm proceeds. How can you change the algorithm to get the reverse thing happening i.e. the partner gets worse and worse for women as the algorithm proceeds?

We know that in the generic G-S Algorithm execution for the Stable Marriage Problem, the partners of men keep getting worse and worse and for women get their partner keeps getting better and better in the case when men do the proposing. Hence we can achieve an algorithm where womens' partner keeps getting worse and worse by having women do the proposing. So if we choose a free woman and then make her decide whom to propose to, we will achieve the required aim.

Q26. What is rank-maximal matching ? Explain with an example.

Rank-maximal allocation is a rule for fair division of indivisible items. For example, considering allocating some items among people. Each person can rank the items based on their liking. According to the RM rule, we have to give as many people as possible their most preferred item, their next-best item to as many people as possible, and so on.

If the case is that each person should receive a single item for example, if we have exactly n different toffies and n children to distribute them among, the problem is called **Rank-Maximal matching**.

Q27. Can we solve rank-maximal matching by using standard G-S algorithm?

In the execution of G-S algorithm, preference lists of both the sets to be matched needs to considered. If an element is already occupied, the G-S algorithm can break that engagement and in favour of a better matching from the perspective of the side receiving the proposals.

However in the case of Rank-Maximal Matching, if an element of the proposal receiving side is already occupied, the proposer would move on to his next preference with no possibility of breaking of this engagement.

Q28. Tim has invited m friends ..... stable matching problem? Justify your answer.

This problem is slightly different from the Stable Marriage Problem in the sense that here, only one set has preferences. Here the two sets to be matched are: (1) The candies. (2) Tim's friends.
But we only have the preference list for each member of the set 'Tim's friends'. There is no preference among 'The candies'. This problem is an example of the Rank-Maximal Matching.