Q1) Consider the following situation where $n = 2$

Members	ordered list
m_1	$\{f_1, f_2\}$
m_2	$\{f_1, f_2\}$
f_1	$\{m_2, m_1\}$
f_2	$\{m_2, m_1\}$

The algo given will output the following valid matching

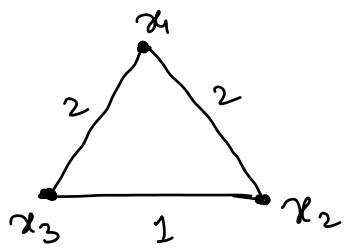
$$\{(m_1, f_1), (m_2, f_2)\}$$

$$\text{Pref}_{f_1}(m_2) > \text{Pref}_{f_1}(m_1)$$

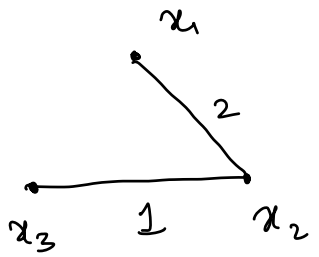
$$\text{Pref}_{m_2}(f_1) > \text{Pref}_{m_2}(f_2)$$

Clearly, the matching is not stable. The given algo does not guarantee a stable matching.

Q2) Consider the following graph

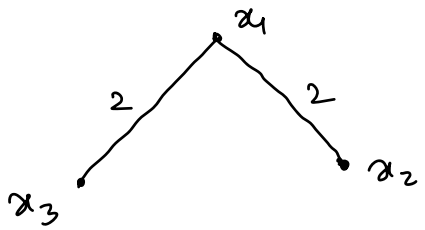


MST rooted at x_1 is the following



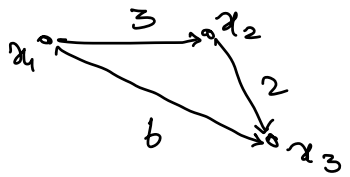
However, the path $x_1 \rightsquigarrow x_3$ in this MST is not the shortest path from x_1 to x_3 in the overall graph.

Q3) Consider the same graph as above. Running Dijkstra's algo starting from x_1, we get



Clearly, this is not the MST.

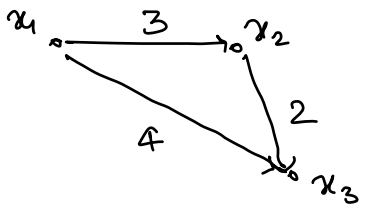
Q4) a) Consider the following graph



Setting the additive factor to > 1, we see that the shortest path between x_1 & x_3 changes.

b) Scaling the edge weights by a constant multiplicative factor does not change the shortest path.

c) Consider the following graph



Clearly, squaring the edge weights changes the shortest path between x_1 & x_3.

Note: Any operation that preserves the ordering of the edge weights will not alter the MST.

$$w_i > w_j \Rightarrow w_i + k > w_j + k$$

$$k \cdot w_i > k \cdot w_j$$

$$w_i^2 > w_j^2$$