In this problem, we need to choose a port among each ship's schedule: the port that the corresponding ship will stay in the rest of the month. Similar to the propose-respond problem discussed in the class, we can simply set up a similar scenario, where ships are regarded as "proposer" while ports are regarded as "respondent". Each ship ranks each port in the chronological order as presented in the schedule; each port ranks each ship in the reverse chronological order as presented in the schedule. According to the class, there will always be a stable matching. Hence, now it is simple to prove such a set of truncations can always be found given the stable matching exists.

Proof

We can use contradictory: A ship s_i gets to port p_i after ship s_j has already staying in p_i. In this case, since s_i prefers p_i to the port in its schedule (because it passes p_i to its destination), and p_i prefers s_i to s_j (rever chronological order), which indicates that s_i should be matched with p_i (i.e., s_j should not be staying there), so it contradicts the stable matching assumption. Hence, we conclude that such a set of truncations can always be found.

The algorithm is stated as follow:

Algorithm 1 Algorithm for problem 3

Regard ships S as "proposers"
Regard ports P as "respondents"
Each ship ranks each port in the chronological order as presented in the schedule
Each port ranks each ship in the reverse chronological order as presented in the schedule
Initially all $s \in S$ and $p \in P$ are unmatched
While there is a ship s that is unmatched and has not "proposed" to every port
    Choose such a ship s
    Let p be the highest-ranked port in s's preference list which s has not yet "proposed"
    If p is not matched then
        (s, p) are matched
    Else p is currently matched to s'
        If p prefers s' to s then
            s remains unmatched
        Else p prefers s to s'
            (s, p) become matched
            s' becomes unmatched
        End if
    End if
End while
Return the set S of matched pairs.

Time complexity analysis:

The complexity of one round is $O(1)$
Observing each ship "proposes" to each port no more than once
n ships $\times$ n ports $= O(n^2)$