CS 4820 HW #1 PROBLEM #3

This problem can be reduced to the stable matching problem as follows:

(1) Use the list of ships as the list of proposers, with their preferences being the ports in the order of visit. That is, if a ship visits port A before port B, we say it "prefers" port A to port B.
(2) Similarly, use the list of ports as the list of respondents, with their preferences being the ships in reverse order of visit. That is, if a port is visited by ship A before ship B, we say it "prefers" ship B to ship A.
(3) Apply the Gale-Shapley algorithm to generate a set of matchings of ships to ports.
(4) For each matching of ship s to port p, stop ship s in port p for maintenance when it arrives there.

This algorithm will generate a list of truncations that satisfies the condition.

Proof. Suppose the algorithm generates a "bad" solution; that is, there exist both a ship s_2 and a matching between ship s_1 and port p_1 such that s_2 stops in p_1 after s_1. This can be reformulated as follows: there exist two matchings, (s_1, p_1) and (s_2, p_2) such that s_2 is set to arrive in p_1 after s_1, and is only due to stop in p_2 after that, causing a conflict as s_2 tries to dock in p_1 that already contains s_1. Since s_2 arrives in p_1 before p_2, in its list of "preferences" as described above, p_1 will come before p_2; similarly, since p_1 is visited first by s_1 and then by s_2, in its list of "preferences", s_2 will come before s_1. Since s_2 is matched to p_2 but prefers p_1, and p_1 is matched to s_1 but prefers s_2, this matching is unstable. Since we have assumed nothing else about the ships' schedule, this implies that if the algorithm generates a "bad" solution, the Gale-Shapley algorithm must have generated an unstable matching. Since the latter is impossible, the algorithm must generate a valid solution. □