(a)

According to the definition of "stable match" described in the class, for any pairs of (r, h):

- (r, h) are matched, which is case 1 in the problem statement
- (r, h') are matched and r prefers h' to h or (r', h) are matched and h prefers r' to r, which is case 2 in the problem statement

Hence, for any set of prospective residents and hospitals and any preference lists, all scenarios can be regarded as the cases of stable matching we learned from the class, which indicates stable matching.

The algorithm is stated as follow:

Algorithm 1 Algorithm for problem 2a

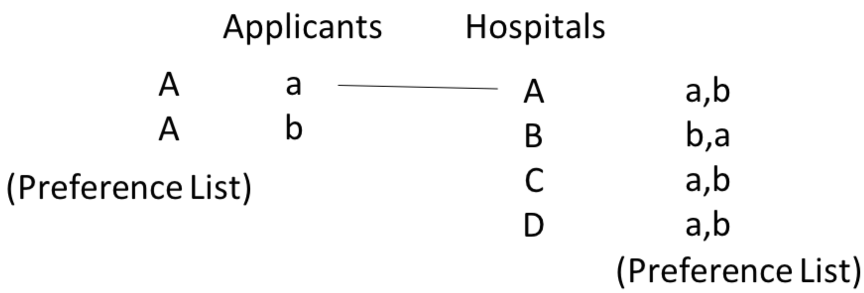
Initially all $r \in R$ have not applied and $h \in H$ have not receiced any applications
While there is an unmatched prospective resident r who has not applied to every hospital in his/her preference list
    Choose such a prospective resident r
    Let h be the highest-ranked hospital in r's preference list which r has not yet applied
    If h is not matched to anyone then
        (r, h) are matched
    Else h is currently matched to r'
        If h prefers r' to r then
            r remains unmatched
        Else h prefers r to r'
            (r, h) become matched
            r' 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 apllicant applies to each hospital no more than once
n applicants $\times$ m hospitals $= O(nm)$

(b)

The answer is **yes**.

Example:



In this scenario, each applicant only has 1 preference (i.e., $k = 1$). Since applicant a, b only prefer A, given the rules, we can only get a matched while b remains unmatched. But if aplicants can apply hospitals other than there top k choices, (i.e., b can apply B, C, D now), then in this case b will be matched to B, which gives a better match.