1. CLR 26-1. (CLRS 25-1).

2. The question refers to the stable matching algorithm as studied in class. We define a pair \((m_i, w_j)\) to be a stable pair if there exists at least one stable marriage at \(m_i\) and \(w_j\) are matched. Prove that during the algorithm studied in class, if \(w_j\) rejects \(m_i\), then they are not a stable pair. Hint — assume this is not the case, consider the first event at which a woman \(w_1\) rejects a man \(m_1\) while \((m_1, w_1)\) are stable pair. Let \(m_2\) be the man whose proposal to \(w_1\) caused her to reject \(m_1\), and show contradiction.

3. Assume that the coin used for setting the level in the SkipList data structure is not biased, and fall in head with probability \(\alpha\), which is not necessarily 1/2. What is the time for inserting an element - and the number of levels it participates in? What is the effect on the query time, and on if \(\alpha\) is much larger than 1/2? What is the effect if it is much smaller?

4. Explain how would you use SkipList to store a set \(S\) of numbers, in a what that would allow you to find the number of elements in \(S\) which are smaller than a query value \(x\). You should be able to add elements to \(S\), delete elements, and perform a query in time \(O(\log |S|)\).

5. Let \(G(V, E)\) denote a graph with weights assign to its edges, where \(V = \{v_1, \ldots, v_n\}\). Read and understand how we can find a matrix \(\Pi[i, j]\), such that \(\Pi[i, j]\) contains the vertex proceeding \(v_i\) on the shortest path from \(v_i\) to \(v_j\).

6. CLR 27.2-4 (CLRS 26.2-4)

7. CLR 27.2-9 (CLRS 26.2-)

8. (only if you feel like it) Let \(L\) be singly connected liked list. \(L\) is called a snake if the last element of \(L\) points to NULL, and is called a snail if the last element points to one of the last elements of \(L\). You are given a pointer to the first
element of $L$, and $O(1)$ additional memory. You cannot change $L$ itself - not even temporarily. Find in $O(n)$ time (a) if $L$ is a snake or a snail, and (b) how many elements are there in $L$. There is a solution that does not use unbounded search.