Fundamental Algorithms, Section 003
Homework 6, Additional Problems, Fall 22.

1. Suppose you are given a collection of DNA strings (they use the 4-character alphabet A,C,G,T). You a writing an algorithm for a genetic engineer who would like to cut and paste from one string to another (no copying of portions of the strings, however; see the challenge problem on the homework for how to support copying). This is defined by specifying the cut as going from the i-th to the j-th characters in one string, and the paste as occurring right after the k-th character in a second string. Show how to maintain the strings so that the cut and paste operations can be performed in $O(\log n)$ assuming the total length of the strings is n.

Present your algorithm so it is clear why it works and justify the runtime briefly. You may state known results about data structures.

2.a. Let S and T be two sets having combined size n.
a. Suppose we were willing to tolerate a 2% incorrect answer rate.
i. Give an algorithm to test if $S \subseteq T$ that runs in worst-case time $O(n)$, but uses only an additional $O(n)$ bits of space. Remember the algorithm is allowed to have a small probability of giving an incorrect answer. Justify your answer.
ii. Why might it be difficult to obtain a similar algorithm to test if S and T are disjoint, again with worst case running time $O(n)$.

b. Suppose instead that we wanted correct answers but would accept an $O(n)$ expected runtime. What are the answers to (i) and (ii) now? Justify your answers.