Problem 1
Problem 4.7 asks to write pseudocode for PartialDigest algorithm that uses fewer lines of code than the one presented in the text. I am not sure what they have in mind, but try to do that. In addition, modify the algorithm to return the first solution found, instead of outputting all of them.

I will skip the solution for this.

Problem 2
Consider the branch-and-bound algorithm for Median String presented in the text on page 114. Assume that instead of finding the median string of length $l$, we want to find all median strings of length 1, 2, $l-1$, $l$. In the process of doing that, use a tighter bound for the branch-and-bound strategy. Hint: Given an $l$-mer $w = uv$, use $\text{TotalDistance}(u, \text{DNA}) + \text{TotalDistance}(v, \text{DNA})$ to bound $\text{TotalDistance}(w, \text{DNA})$.

Solution: We know that if $w = uv$, then

$$\text{TotalDistance}(u, \text{DNA}) + \text{TotalDistance}(v, \text{DNA}) \leq \text{TotalDistance}(w, \text{DNA})$$

In the algorithm on page 114, the optimistic distance is calculated as $\text{TotalDistance}(\text{prefix}, \text{DNA})$ where prefix has length $i$. However, the optimistic distance should actually be larger. Given $l$, we would have solved the problem for all lengths up to $l - 1$. Therefore, given the best word of length $l - i$, and let $D$ be its distance, one could consider $\text{TotalDistance}(\text{prefix}, \text{DNA}) + D$ knowing that this is a lower bound on the actual distance for any word of length $l$. This is a tighter bound for the optimistic distance.

Problem 3
(a) We saw in class that given a permutation of numbers from 1 to $n$ (with 0 and $n+1$ fixed on left and right respectively), if there is a decreasing strip, then there is a reversal that reduces the number of breakpoints. This means that if no reversal that reduces the number of breakpoints exists, then all strips must be increasing. Show by example that the converse is not true, i.e. it’s possible that all strips are increasing and yet there is a reversal that reduces the number of breakpoints.

Solution: Here’s an example with 3 breakpoints:
The reversal of 523 will result in 2 breakpoints.

(b) How many permutations have one breakpoint?

Solution: zero. There cannot be permutations with only 1 breakpoint.

(c) How many permutations have two breakpoints?

Solution: Let the breakpoints be numbered 0 to n, where breakpoint $i$ sits between the $i^{th}$ and $(i+1)^{st}$ position (so breakpoint 0 is at the beginning and breakpoint $n$ is at the end). Every pair of breakpoints give a valid scenario, except if the two breakpoints are consecutive (which will result in the same permutation). Therefore, we have $n(n+1)/2 - n$ possible pairs.

Problem 4

Do problem 8.6.

Solution: we have \{AGT, AAA, ACT, AAC, CTT, GTA, TTT, TAA\}. A shortest common superstring is: AGTAAACTTT. We know this is the shortest because it starts with AGT and then uses one character per every additional 3-mer. This can be easily retrieved using the Euler path technique.