INSTRUCTIONS: Please review carefully the instructions given for Homework 1. They apply to this assignment, too.

Please hand in your answers to the following problems. Problem numbers, where indicated, are from the seventh edition of the Rosen text.

1. (4.5 points) p. 176, #2, (a), (d), (e). If a set is finite or uncountable, then just state so. If it is countably infinite, then state so and give a bijective function (you do not have to prove that it is a bijection).

2. (4.5 points) p. 176, #10. Justify your answer for each part briefly.

3. (4 points) Let $A$ and $B$ be infinite sets with the same cardinality. Prove that $\mathcal{P}(A)$ and $\mathcal{P}(B)$ also have the same cardinality. Do this by giving explicitly a bijective function from $\mathcal{P}(A)$ to $\mathcal{P}(B)$. You must also prove that your function is indeed a bijection.

4. (4 points) p. 177, #22.

5. (4 points) Use diagonalization to prove that the set consisting of all infinite-length binary strings is uncountable. Be sure to spell out the steps in your proof carefully.

6. (6 points) Let $T = t_1 t_2 \ldots t_n$ and $P = p_1 p_2 \ldots p_m (m \leq n)$ be two text strings composed of letters from (say) the English alphabet. We wish to decide whether $P$ is a substring of $T$, i.e., whether the letters of $P$ appear consecutively in $T$. (For instance, if $T$ is MINNESOTA and $P$ is INNE, then $P$ is a substring; if $P$ is INEN, then it is not.) This is a common task in text editors. Your goal is to develop a simple algorithm that outputs “yes”/“no” depending on whether or not $P$ is a substring of $T$. Your algorithm should make no more than $m \times n$ comparisons between letters from $P$ and $T$.

   (a) Describe your approach in a few sentences and give pseudocode. Your style of pseudocode should be similar to the one in the text or seen in class; do not give full-blown C/C++/Java etc.

   (b) Analyse your algorithm and show that it makes no more than the stated number of comparisons between letters of the two strings. (Ignore any other type of operation.)

7. (8 points) Let $A[1 : n]$ be an array that contains a sequence of distinct integers (in some arbitrary order). We wish to find both the minimum and the maximum element in $A$. 

Over
(a) Give an algorithm for this problem that makes just one pass over \( A \) and determines the minimum and maximum using a total of \( 2n - 2 \) comparisons between elements of \( A \). (Ignore any other type of operation.) Describe your approach in a few sentences, give pseudocode, and establish the stated bound on the number of comparisons made by your algorithm. (See Problem 6’s statement for the style of pseudocode to use.)

(b) Suppose that you are allowed to make more than one pass over \( A \). Show how to find both the minimum and the maximum using just \( 3(n/2) - 2 \) element comparisons. (For simplicity, assume that \( n \) is even here.) You may explain your approach in words (pseudocode is not required). Be sure to establish the stated bound on the number of comparisons.