Question 1. Consider an example of the turnpike problem where the multiset $L$ is \{2, 3, 4, 4, 5, 5, 7, 7, 9, 9, 10, 11, 14, 14, 18\}. Solve the turnpike problem for $L$ using an “intelligent” exhaustive search algorithm. Give a brief description of how your exhaustive search algorithm works.

Answer. Using the knowledge that each of the distances from the beginning of the turnpike to the exits is an element in $L$, one exhaustive search algorithm is as follows:

1. compute the number of exits in the turnpike, $n$
2. iterate over each permutation of elements of $L$, call it $\pi$, of size $n - 1$ (also append 0 to the beginning of $\pi$, as the distance to the initial exit)
3. check if $\pi$ generates the distances in $L$ (for example, by computing the distances between each of the $\binom{n}{2}$ pairs of numbers in $\pi$)

By the assumption stated above, one of the permutations $\pi$ must be a solution, so this algorithm will find it eventually, by exhausting all possible turnpike layouts.

Question 2. (This problem is a small lesson in reading the textbook.) Solve the turnpike problem for $L$ in problem 1 using the “branch and bound” version of the algorithm given in Section 4.3 on page 89 of the textbook.

Answer. Since $\binom{n}{2} = 15$, $n$ must equal 6. Call the distances at the 6 exits $x_1, x_2, \ldots, x_6$ in the solution $X$. We know $x_1 = 0$, and the largest element of $L$ is 18, so $x_6 = 18$. After removing 18 from $L$ we have

$$X = \{0, 18\} \quad L = \{2, 3, 4, 4, 5, 5, 7, 7, 9, 9, 10, 11, 14, 14\}.$$

The largest remaining distance is 14, so either $x_5 = 14$ or $x_2 = 18 - 14 = 4$. Since these two possibilities result in mirror image turnpikes, We will let $x_2 = 4$. After removing $x_6 - x_2 = 14$ and $x_2 - x_1 = 4$ from $L$, we have

$$X = \{0, 4, 18\} \quad L = \{2, 3, 4, 5, 5, 7, 7, 9, 9, 10, 11, 14\}.$$
The largest remaining distance is 14, so either \( x_3 = 4 \) or \( x_5 = 14 \). If \( x_3 = 4 \), then \( x_3 - x_2 = 0 \), which is not in \( L \), so it must be the case that \( x_5 = 14 \). After removing \( x_3 - x_5 = 4 \), \( x_5 - x_2 = 10 \), and \( x_5 - x_1 = 14 \) from \( L \), we have
\[
X = \{0, 4, 14, 18\} \quad L = \{2, 3, 5, 7, 7, 9, 9, 11\}.
\]
The largest remaining distance is 11, so either \( x_3 = 7 \) or \( x_4 = 7 \). If \( x_3 = 7 \), the distances \( x_6 - x_3 = 11 \), \( x_5 - x_3 = 7 \), \( x_3 - x_2 = 3 \), and \( x_3 - x_1 = 7 \) are all elements in \( L \), so this setting of \( x_3 \) is valid. After removing those elements from \( L \), we have
\[
X = \{0, 4, 7, 14, 18\} \quad L = \{2, 5, 5, 9, 9\}.
\]
The largest remaining distance is 9 and the only unset variable is \( x_4 \), so \( x_4 = 9 \), yielding the solution \( X = \{0, 4, 7, 9, 14, 18\} \). We can verify from \( X \) that all \( \binom{3}{2} \) distances between elements in \( X \) are the same numbers from the multiset \( L \), so \( X \) is a correct solution to this instance of the turnpike problem. \(\square\)

**Question 3** (page 119, problem 4.8). Find a set \( \Delta X \) with the smallest number of elements that could have arisen from more than one \( X \), not counting shifts and reflections.

**Remark.** This question was not graded.

**Answer.** Page 87 of the textbook provides an example with a \( \Delta X \) of size 36, so the sets from which it arises are of size 9. An example with \( \Delta X \) of size 15 arises from \( X = \{0, 1, 2, 5, 7, 9, 12\} \) and \( X' = \{0, 1, 5, 7, 8, 10, 12\} \), where the \( \Delta X = \Delta X' = \{1, 1, 2, 2, 2, 3, 3, 4, 4, 5, 5, 5, 6, 7, 7, 7, 8, 8, 9, 9, 11, 12\} \). Here \( |X| = |X'| = 6 \). Now we will consider sets of size less than 6.

A \( \Delta X \) with a single element \( x \) has only one solution, \( X = \{0, x\} \), if we ignore its shifts and reflections.

The next possible size for \( \Delta X \) is 3, since the size must be of the form \( \binom{n}{2} \), and \( \binom{3}{2} = 3 \). Let \( \Delta X \) be \( \{a, b, c\} \), where \( a \leq b \leq c \). In any case, the first exit must be at distance 0 and the last exit at distance \( c \). Therefore, for \( \Delta X \) to be valid it must be true that \( c - b = a \), and \( c - a = b \), and the exits are also at those distances, so flipping the values of \( a \) and \( b \) result in the same output. This means that any \( \Delta X \) of size 3 produces two turnpikes, each of which is the reflection of the other.

Remaining possibilities for the size of \( \Delta X \) is \( \binom{4}{2} = 6 \) and \( \binom{5}{2} = 10 \). We are not sure if there are \( \Delta X \) of size 6 or 10. \(\square\)

**Question 4** (page 54, problem 2.1). Write an algorithm that given a list of \( n \) numbers, returns the largest and smallest numbers in the list. Estimate the running time of the algorithm. Can you design an algorithm that performs only \( 3n/2 \) comparisons to find the smallest and largest numbers in the list?

**Answer.** Let \( L = (x_1, x_2, \ldots, x_n) \) be the list to sort. Assume for simplicity that \( n \) is even. Create two initially empty lists \( L_{\text{small}} \) and \( L_{\text{large}} \). For \( i = 0 \) up to \( n/2 \), compare \( x_i \) to \( x_{i+1} \) and put the smaller into \( L_{\text{small}} \) and the larger into
Finally, do a linear search through $L_{\text{small}}$ to find the smallest element, and through $L_{\text{large}}$ to find the largest element.

This algorithm requires $n/2$ comparisons to split the pairs of $L$ into $L_{\text{small}}$ and $L_{\text{large}}$, $n/2 - 1$ comparisons to find the smallest element in $L_{\text{small}}$, and $n/2 - 1$ comparisons to find the largest element in $L_{\text{large}}$. Therefore the total number of steps is $3n/2 - 2$.

**Example.** Let $L = (6, 2, 5, 3, 10, 5, 3, 4, 7)$. We make 5 pairwise comparisons to split the list into $L_{\text{small}} = (2, 3, 5, 3, 4)$ and $L_{\text{large}} = (6, 5, 10, 3, 7)$. To find the smallest element of $L_{\text{small}}$ we must compare the first element to each other, so that is 4 comparisons. To find the largest element of $L_{\text{large}}$ we must compare the first element to each other, so that is also 4 comparisons. Therefore the total number of comparisons is $5 + 4 + 4 = 13$, which is $3(10)/2 - 2$ as expected.

**Question 5.** You are given a set of intervals, $X$, each of which starts at some non-negative integer and ends at some larger integer. A subset $Y$ of these intervals is called a covering if any (real) number in one of the intervals in $X$ is also in some interval in $Y$. The size of $Y$ is the number of intervals in $Y$.

Give a greedy algorithm for finding the covering of $X$ with the smallest size.

Show how your example works on a set $X$ of 8 intervals, where $X = \{[1, 3], [0, 3], [7, 10], [0, 5], [3, 5], [6, 8], [2, 3], [6, 10]\}$.

Explain why your algorithm always gives an optimal covering $Y$, that is a smallest possible covering.

**Answer.** First we define a few terms. The *size* of an interval $[a, b]$ is its length, namely $b - a$. For any number $c$ in the interval $[a, b]$, the *size of the interval to the right of $c$* is the size of the interval $[b, c]$. An optimal or minimal cover of $X$ is a cover which contains the fewest possible intervals from $X$. Such a cover may not be unique.

One greedy algorithm which computes a minimal cover of $X$ is as follows.

We will construct a set $M$ containing intervals from $X$ which will cover $X$. First compute the set $U$ which is the union of the intervals in $X$. Note that $U$ is itself a finite collection of intervals. Repeat the following step until all of $U$ is covered:

1. Let $\text{max}$ be the largest number in our set $M$ so far. (By this we mean $\text{max}$ is a member of one of the intervals in $M$.) Initially $M = \emptyset$ and $\text{max} = 0$.

2. Let $\text{new}$ be the smallest number in $U$ not yet contained in an interval in $M$. If there is no such number, $U$ is covered so we halt and output $M$.

3. Find the interval $I$ in $X$ which contains $\text{new}$ and for which the size of $I$ to the right of $\text{new}$ is largest.

4. Add $I$ to $M$.

This concludes the algorithm.
Example. Suppose \( X = \{[1,3], [0,3], [7,10], [0,5], [3,5], [6,8], [2,3], [6,10]\} \). Initially \( \text{max} = 0 \). Also, \( \text{new} \) is 0, since 0 is the least number in \( U \) not yet in \( M \).

Our algorithm now chooses the interval \([0,5]\) and puts it into \( M \), since the size of \([0,5]\) to the right of \( \text{new} \) is 5, which is the largest possible size.

Now we iterate this step again. At this point \( \text{max} \) is 5, and \( \text{new} \) is 6. We choose the interval \([6,10]\).

As these two intervals together cover all of \( X \), we halt when we reach step 2. We output \( M = \{[0,5], [6,10]\} \). \( M \) has size 2 which is optimal.

Now to prove our algorithm works and the \( M \) which is constructed contains as few intervals from \( X \) as possible, consider every initial segment \((M_1, M_2, \ldots, M_k)\) of \( M = \{M_1, M_2, \ldots, M_t\} \) which we construct in a run of the algorithm. We will prove by induction that each such initial segment is the subset of some minimal cover. From this claim it follows that the whole collection \( M = \{M_1, M_2, \ldots, M_t\} \) is the subset of some minimal cover, say \( M^* \). But since this collection is itself a cover of \( X \), \( M^* \), being optimal, cannot contain any more intervals than are in \( M \), and so any optimal cover has \( t \) intervals in it. Hence \( M \) is one such cover, and so it is optimal.

Claim. Every initial segment \((M_1, M_2, \ldots, M_k)\) of \( M \) is contained in some optimal cover.

Proof. The proof is by induction on the number of sets in the initial segment.

Base case. \( M_1 \) is contained in some optimal cover of \( X \).

Proof. Let \( Y \) be any optimal cover of \( X \), and assume \( M_1 \) is not in \( Y \). (If it is, there is nothing to prove.) Let \( a \) be the smallest number in any interval in \( X \). Choose an interval \( I \) from \( Y \) which contains \( a \). Then replace \( I \) with \( M_1 \) in \( Y \). Since \( M_1 \) is chosen by the algorithm to be as large as possible, this updated \( Y \) is still an optimal cover, and it now contains \( M_1 \). \( \square \)

Inductive step. If there is an optimal cover \( Y \) of \( X \) which contains each of \( M_1, M_2, \ldots, M_i \), then there is an optimal cover \( Z \) of \( X \) which contains \( M_1, M_2, \ldots, M_i, M_{i+1} \).

Proof. Let \( a \) be the least element in any interval in \( X \) which is not in the union of \( M_1, M_2, \ldots, M_i \). Then by our construction, \( M_{i+1} \) is the interval \( I \) in \( X \) which contains \( a \) and for which the size of \( I \) to the right of \( a \) is largest.

Let \( I' \) be any interval in \( Y \) containing \( a \). Let \( Z \) be \( Y \) with \( I' \) replaced with \( M_{i+1} \). Then \( Z \) is still a cover of \( X \), it is optimal since it has the same number of intervals as \( Y \), and it contains each of \( M_1, M_2, \ldots, M_i, M_{i+1} \). \( \square \)