2-4 Inversions
Let \( A[1..n] \) be an array of \( n \) distinct numbers. If \( i < j \) and \( A[i] > A[j] \), then the pair \((i, j)\) is called an inversion of \( A \).

\( a. \) List the five inversions of the array \( (2, 3, 8, 6, 1) \).

\( b. \) What array with elements from the set \( \{1, 2, \ldots, n\} \) has the most inversions? How many does it have?

\( c. \) What is the relationship between the running time of insertion sort and the number of inversions in the input array? Justify your answer.

\( d. \) Give an algorithm that determines the number of inversions in any permutation on \( n \) elements in \( \Theta(n \log n) \) worst-case time. (Hint: Modify merge sort.)

2.2) Given the following:
\[
T(n) = \begin{cases} 
T\left(\left\lfloor \frac{n}{2} \right\rfloor \right) & + T\left(\left\lfloor \frac{n}{2} \right\rfloor \right) + n - 1 & \text{if } n > 1 \\
0 & & \text{if } n = 1
\end{cases}
\]

Show by mathematical induction the solution to the above equation is:
\[
T(n) = n \left\lfloor \log_2 n \right\rfloor - 2 \left\lfloor \frac{\log_2 n}{2} \right\rfloor + 1
\]

3) In the algorithm SELECT, the input elements are divided into groups of 5. Will the algorithm work in linear time if they are divided into groups of 7? Argue that SELECT does not run in linear time if groups of 3 are used.