1. We order an array of \( n \) numbers in increasing order using insertion sort. What is the running time of the algorithm in the following cases?

   (a) The array contains \( n \) different numbers and they are given in increasing order.
   (b) The array contains \( n \) different numbers and they are given in descending order.
   (c) The array consists of the same number \( n \) times.

2. What are the answers if we change the sorting method in the previous exercise to heapsort?

3. What are the answers if we change the sorting method in the previous exercise to mergesort?

4. What are the answers if we change the sorting method in the previous exercise to quicksort using the \( \textsc{Partition} \) function used in the lectures (the pivot is the first number in the array, not last as in Cormen p. 146)?

5. What would happen, if \( \textsc{Partition}(A, p, r) \) used in the lectures returned \( r \)? Show that the \( \textsc{Partition}(A, p, r) \) function used in the lectures never returns \( r \).

6. Suppose arrays \( A \) and \( B \) are both sorted in increasing order and both contain \( n \) numbers. Give an \( O(n \log n) \) algorithm to find the median of the combined array \( A \cup B \).

   Note: The median is the middle number, when the numbers are sorted in order. If there is an even number of numbers, it is the mean of the two numbers in the middle.