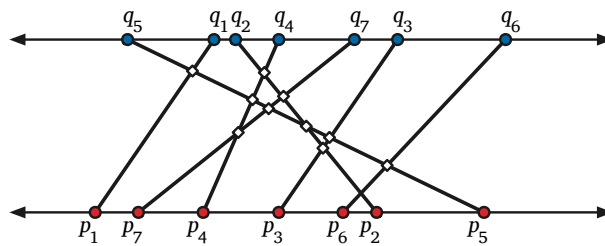


1. An *inversion* in an array  $A[1..n]$  is a pair of indices  $(i, j)$  such that  $i < j$  and  $A[i] > A[j]$ . The number of inversions in an  $n$ -element array is between 0 (if the array is sorted) and  $\binom{n}{2}$  (if the array is sorted backward).

Describe and analyze a divide-and-conquer algorithm to count the number of inversions in an  $n$ -element array in  $O(n \log n)$  time. Assume all the elements of the input array are distinct.

2. Suppose you are given two sets of  $n$  points, one set  $\{p_1, p_2, \dots, p_n\}$  on the line  $y = 0$  and the other set  $\{q_1, q_2, \dots, q_n\}$  on the line  $y = 1$ . Create a set of  $n$  line segments by connect each point  $p_i$  to the corresponding point  $q_i$ . Describe and analyze a divide-and-conquer algorithm to determine how many pairs of these line segments intersect, in  $O(n \log n)$  time. [Hint: Use your solution to problem 1.]

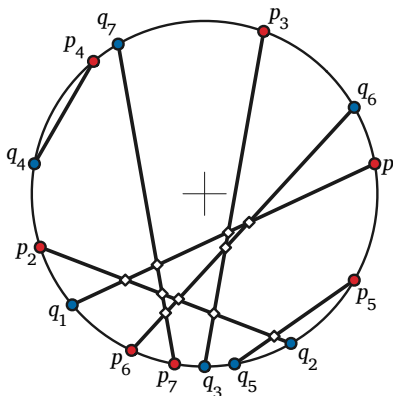
Assume a reasonable representation for the input points, and assume the  $x$ -coordinates of the input points are distinct. For example, for the input shown below, your algorithm should return the number 10.



Ten intersecting pairs of segments with endpoints on parallel lines.

3. Now suppose you are given two sets  $\{p_1, p_2, \dots, p_n\}$  and  $\{q_1, q_2, \dots, q_n\}$  of  $n$  points *on the unit circle*. Connect each point  $p_i$  to the corresponding point  $q_i$ . Describe and analyze a divide-and-conquer algorithm to determine how many pairs of these line segments intersect in  $O(n \log^2 n)$  time. [Hint: Use your solution to problem 2.]

Assume a reasonable representation for the input points, and assume all input points are distinct. For example, for the input shown below, your algorithm should return the number 10.



Ten intersecting pairs of segments with endpoints on a circle.

4. **To think about later:** Solve problem 3 in  $O(n \log n)$  time.