We'll define a recursive divide-and-conquer algorithm ALG which takes a sequence of distinct numbers $a_1, \ldots, a_n$ and returns N and $a'_1, \ldots, a'_n$ where

- N is the number of significant inversions
- $a'_1, \ldots, a'_n$ is same sequence sorted in the increasing order

ALG is similar to the algorithm from the chapter that computes the number of inversions. The difference is that in the 'conquer' step we merge twice: first we merge $b_1, \ldots, b_k$ with $b_{k+1}, \ldots, b_n$ just for sorting, and then we merge $b_1, \ldots, b_k$ with $2b_{k+1}, \ldots, 2b_n$ for counting significant inversions.

Let's define ALG formally. For $n = 1$ ALG just returns $N = 0$ and $\{a_1\}$ for the sequence. For $n > 1$ ALG does the following:

- let $k = \lfloor n/2 \rfloor$.
- call ALG$(a'_1, \ldots, a'_k)$. Say it returns N_1 and $b_1, \ldots, b_k$.
- call ALG$(a'_{k+1}, \ldots, a'_n)$. Say it returns N_2 and $b_{k+1}, \ldots, b_n$.
- compute the number N_3 of significant inversions (a_i, a_j) where $i \leq k < j$.
- return $N = N_1 + N_2 + N_3$ and $a'_1, \ldots, a'_n = \mathsf{MERGE}(b_1, \ldots, b_k;\ b_{k+1}, \ldots, b_n)$

MERGE can be implemented in $O(n)$ time. According to the discussion in the book, it remains to find a way to compute N_3 in $O(n)$ time. We implement a variant of merge-count of $b_1, \ldots, b_k$ and $2\,b_{k+1}, \ldots, 2\,b_n$ as follows.

- Initialize counters: $i \leftarrow k$, $j \leftarrow n$, $N_3 \leftarrow 0$.
- If $b_i \leq 2b_j$ then
  - if $j > k + 1$ decrease j by 1.
  - if $j = k + 1$ return N_3.
- If $b_i > 2b_j$ then increase N_3 by $j - k$. Then
  - if $i > 1$ decrease i by 1.
  - if $i = 1$ return N_3.

Explanation For every i we count the number of significant inversions between b_i and all b_j's. If $b_i \leq 2b_j$ then there are no significant inversions between b_i and any b_m s.t. $m \geq j$, so we decrease j. If $b_i > 2b_j$ then $b_i > 2b_m$ for all m s.t. $k < m \leq j$. In other words, we have detected $j - k$ significant inversions involving b_i. So we increase N_3 by $j - k$. Finally, when we are down to $i = 1$ and have counted significant inversions involving b_1, there are no more significant inversions to be detected.

[1] ex499.218.598