Let $A[1..n]$ be an array of n distinct and positive integers. Given an integer x, describe an $O(n \log n)$ algorithm to return the number of distinct pairs (i, j) of indices where $1 \leq i < j \leq n$, satisfying:

- $A[i] > A[j]$, and
- $A[i] + A[j] = x$.

Note. *This was an assignment problem from COMP3121/9101, 23T2.*

Rubric.

- This task will form part of the portfolio.
- Ensure that your argument is clear and keep reworking your solutions until your lab demonstrator is happy with your work.

Solution. First, sort the array A along with their original indices. We will call the sorted array B. This can be done in $O(n \log n)$ time using merge sort. For each element $B[i]$ in B, perform a binary search to find the index j such that $B[j] = x - B[i]$ and $B[j]$'s original index is less than $B[i]$'s original index. Note that the elements in A are distinct so there will not be any double-ups. This ensures that we only count pairs that satisfy both conditions: $A[i] + A[j] = x$ and $A[i] > A[j]$. During the binary search, keep a count of the number of valid pairs found. This count will be the final answer. The binary search for each element takes $O(\log n)$ time, so this step takes a total of $O(n \log n)$ time.