Complexity Analysis:

Divide:

We divide the process into two equal subarray. Thus if the original program takes T(n) time then, the subdivided problem takes T(n/2) amount of time.

Conquer:

In the conquer step we find the smallest distance between two subarray, this takes O(1) time

Merge:

In the merge step we determine the smallest interval between two arrays. This needs sorting and a linear time searching. That means the complexity of this step is O(n ln n) + O(n) or O(n lg n)

That means, the complexity of searching the smallest distance is T(n) = 2T(n/2) + O(n lg n)

If we take the substitution approach then substituting the complexity O(n lg n)

We can see than T(n) = O(n lg n)

In the search of second smallest distance we used the algorithm three times, that means our time complexity becomes O(n lg n)