

**Question 1. Pair Sums** (35 marks)

The class [SortedIntegerArray](#) contains a sorted representation of an array of integers. You are to implement a method **kPairSum**, that uses a **recursive** algorithm to determine whether the array contains two elements that sum to a given integer  $k$ , returning true if it does, false if it does not. **The method should run in  $O(n)$  time**, where  $n$  is the number of integers in the array.

For example, given the array [3 4 4 6 8 9], **kPairSum**(10) should return true, and **kPairSum**(16) should return false.

Be careful about overflow and underflow errors. For example, what does your program do if the array contains two elements that are both equal to Integer.MAX\_VALUE? **Hint 1:** You can deal with this by representing the sum with a variable of type long.

**Hint 2:** the method **kPairSum** does not have to be recursive itself. Instead, create a private recursive method within SortedIntegerArray private boolean **kPairSumInterval**(Integer  $k$ , int  $i$ , int  $j$ ) that solves the problem for the subinterval of the array from index  $i$  to  $j$ . Now call this method from **kPairSum**.

Here is a test program [testSortedIntegerArray](#), that provides random test cases. You should, however, test your program using a broader range of test cases. Pay particular attention to boundary conditions.