Homework Sheet 2

Authors: Abdullah Oğuz Topçuoğlu & Ahmed Waleed Ahmed Badawy Shora & Yousef Mostafa Farouk

Exercise 1

Given an array A[1, . . . , n] of integers and an integer k, we can decide whether there are two distinct entries in A that are k-similar by following these steps:

1. Sort the array A in ascending order. This can be done in O(n log n) time using merge sort from the lecture.
2. After sorting we are gonna start from the beginning of the sorted array and compare each element with the next element to check if their difference is less than or equal to k and greater than zero (greater than zero because elements needs to be distinct).

Pseudocode:

```
procedure AreKSimilar(A[1...n], k)
   MergeSort(A)  // from the lecture, O(n log n) time
   for i = 1 to n-1 do
      if A[i+1] - A[i] <= k and A[i+1] != A[i] then
         return True
   return False
```

Correctness:
After sorting the array, all elements that are k-similar will be positioned next to each other. By iterating through the sorted array and checking the difference between consecutive elements, we can determine if there are any two distinct entries that are k-similar. If we find such a pair we return true, otherwise, we return false after checking all pairs.
Running Time Analysis:
The sorting step takes O(n log n) time. The subsequent loop iterates through the array once, taking O(n) time. Therefore, the overall time complexity of the algorithm is O(n log n) + O(n) = O(n log n).