Given an array of **2n** integers, your task is to group these integers into **n** pairs of integer, say (a1, b1), (a2, b2), ..., (an, bn) which makes sum of min(ai, bi) for all i from 1 to n as large as possible.

**Example 1:**

**Input:** [1,4,3,2]

**Output:** 4

**Explanation:** n is 2, and the maximum sum of pairs is 4.

**Note:**

1. **n** is a positive integer, which is in the range of [1, 10000].
2. All the integers in the array will be in the range of [-10000, 10000].

**Solution:**  Intuitive solution is to sort the array, and club two highest numbers (ai, bi). The smallest among them is used for calculating the sum. This is a greedy algorithm. Now, since the numbers in the list are only between -10000 to +10000, we can sort them using an array (index them using the numbers themselves). Then, the highest index, in the array whose indexed value is not zero, is the biggest number in the list. Though it takes O(range) memory, we’re getting the solution in O(n) time.