Problem 1099: Two Sum Less Than K

Problem Information

Difficulty: Easy
Acceptance Rate: 62.09%
Paid Only: Yes
Tags: Array, Two Pointers, Binary Search, Sorting

Problem Description

Given an array `nums` of integers and integer `k`, return the maximum `sum` such that there exists `i < j` with `nums[i] + nums[j] = sum` and `sum < k`. If no `i`, `j` exist satisfying this equation, return `-1`.

Example 1:

Input: nums = [34,23,1,24,75,33,54,8], k = 60 **Output:** 58 **Explanation:** We can use 34 and 24 to sum 58 which is less than 60.

Example 2:

Input: nums = [10,20,30], k = 15 **Output:** -1 **Explanation:** In this case it is not possible to get a pair sum less that 15.

Constraints:

* `1 <= nums.length <= 100` * `1 <= nums[i] <= 1000` * `1 <= k <= 2000`

Code Snippets

C++:

```
class Solution {
public:
int twoSumLessThanK(vector<int>& nums, int k) {
```

```
}
};
```

Java:

```
class Solution {
public int twoSumLessThanK(int[] nums, int k) {

}
}
```

Python3:

```
class Solution:
def twoSumLessThanK(self, nums: List[int], k: int) -> int:
```