881. Boats to Save People

You are given an array `people` where `people[i]` is the weight of the `i<sup>th</sup>` person, and an **infinite number of boats** where each boat can carry a maximum weight of `limit`. Each boat carries at most two people at the same time, provided the sum of the weight of those people is at most `limit`.

Return *the minimum number of boats to carry every given person*.

Example 1:

```
Input: people = [1,2], limit = 3
Output: 1
Explanation: 1 boat (1, 2)
```

Example 2:

```
Input: people = [3,2,2,1], limit = 3
Output: 3
Explanation: 3 boats (1, 2), (2) and (3)
```

Example 3:

```
Input: people = [3,5,3,4], limit = 5
Output: 4
Explanation: 4 boats (3), (3), (4), (5)
```

Constraints:

- `1 <= people.length <= 5 * 10<sup>4</sup>`
- `1 <= people[i] <= limit <= 3 * 10<sup>4</sup>`
- class Solution: def numRescueBoats(self, people: List[int], limit: int) -> int: people.sort() boats = 0 i = 0 j = len(people)-1 while i<=j: temp = people[i]+people[j] if i!=j else people[i] if temp<=limit: i = i+1 j = j-1 boats+=1 elif temp>limit: j = j-1 boats+=1 return boats

  ```
  class Solution:
      def numRescueBoats(self, people: List[int], limit: int) -> int:
          people.sort()
          boats = 0
          i = 0
          j = len(people)-1
          while i<=j:
              temp = people[i]+people[j] if i!=j else people[i]
  ```

```
            if temp<=limit:
                i = i+1
                j = j-1
                boats+=1
            elif temp>limit:
                j = j-1
                boats+=1
        return boats
```