# Count of subsets having sum of min and max element less than K

# Problem Statement

## **For a given vector of integers and integer K, find the number of non-empty subsets S such that min(S) + max(S) <= K. For example, for K = 8 and vector [2, 4, 5, 7], the solution is 5 ([2], [4], [2, 4], [2, 4, 5], [2, 5]).**

# Explanation & Algorithm

### Sort the input array first.

### Now use Two Pointer Technique to count the number of subsets.

### Let take two pointers left and right and set left = 0 and right = N-1.

### Check for below conditions

### if (arr[left] + arr[right] <= K ). Increment the left pointer by 1 and add 2^(j – i) into answer, because the left and right values make up a potential end values of a subset. All the values from [i, j – 1] also make up end of subsets which will have the sum < K. So, we need to calculate all the possible subsets for left = i and right ∊ [i, j]. So, after suming up values 2^(j – i + 1) + 2^(j – i – 2) + … + 2 0 of the GP, we get 2^(j – i) .

### if( arr[left] + arr[right] > K )

### Decrement the right pointer by 1.

### Repeat the below process until left <= right.

# Time Complexities

O(N\*log(N))

* Sorting (n) \* log(n) to find subset for each number pair

# Test Cases

## Test Case 1

Input: arr[] = {2, 4, 5, 7} K = 8

Output: 5

Subsets: ([2], [4], [2, 4], [2, 4, 5], [2, 5])

## Test Case 2

Input: arr[] = {2, 2, 5, 6} K = 7

Output: 5

Subsets: ([2], [2], [2, 2], [2, 5], [2, 5], [2, 2, 5])

## Test Case 3

Input:: arr[] = {2, 4, 2, 5, 7} K = 10

Output: 27