Problem 1755: Closest Subsequence Sum

Problem Information

Difficulty: Hard
Acceptance Rate: 42.63%
Paid Only: No
Tags: Array, Two Pointers, Dynamic Programming, Bit Manipulation, Sorting, Bitmask

Problem Description

You are given an integer array `nums` and an integer `goal`.

You want to choose a subsequence of `nums` such that the sum of its elements is the closest possible to `goal`. That is, if the sum of the subsequence's elements is `sum`, then you want to **minimize the absolute difference** `abs(sum - goal)`.

Return _the**minimum** possible value of_ `abs(sum - goal)`.

Note that a subsequence of an array is an array formed by removing some elements **(possibly all or none)** of the original array.

Example 1:

Input: nums = [5,-7,3,5], goal = 6 **Output:** 0 **Explanation:** Choose the whole array as a subsequence, with a sum of 6. This is equal to the goal, so the absolute difference is 0.

Example 2:

Input: nums = [7,-9,15,-2], goal = -5 **Output:** 1 **Explanation:** Choose the subsequence [7,-9,-2], with a sum of -4. The absolute difference is abs(-4 - (-5)) = abs(1) = 1, which is the minimum.

Example 3:

Input: nums = [1,2,3], goal = -7 **Output:** 7

Constraints:

* `1 <= nums.length <= 40` * `-107 <= nums[i] <= 107` * `-109 <= goal <= 109`

Code Snippets

C++:

```
class Solution {
public:
int minAbsDifference(vector<int>& nums, int goal) {

}
};
```

Java:

```
class Solution {
public int minAbsDifference(int[] nums, int goal) {

}
}
```

Python3:

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