The Capacity to Ship Packages within D Days LeetCode

You have a conveyor belt with packages that need to be shipped from one port to another within a certain number of days. Each package on the conveyor belt has a specific weight, given by the array 'weights'. You need to determine the minimum weight capacity of a ship that can ship all the packages within the given number of days.

The goal is to find the least weight capacity of the ship that will result in all the packages being shipped within 'days' days, while maintaining the order of the packages on the conveyor belt.

Examples:

Input: weights = [1,2,3,4,5,6,7,8,9,10], days = 5

Output: 15

Explanation: In this example, we can ship all the packages in 5 days by using a ship capacity of 15. The packages can be loaded onto the ship as follows:

1st day: 1, 2, 3, 4, 5

2nd day: 6, 7

3rd day: 8

4th day: 9

5th day: 10

Note that the packages must be loaded in the given order, so splitting them into different parts on the same day is not allowed.

Input: weights = [3,2,2,4,1,4], days = 3

Output: 6

Explanation: In this example, a ship capacity of 6 is the minimum required to ship all the packages in 3 days. The packages can be loaded as follows:

1st day: 3, 2

2nd day: 2, 4

3rd day: 1, 4

Input: weights = [1,2,3,1,1], days = 4

Output: 3

Explanation: Here, a ship capacity of 3 is sufficient to ship all the packages in 4 days while maintaining their order.

1st day: 1

2nd day: 2

3rd day: 3

4th day: 1, 1

The task is to determine the minimum weight capacity of the ship that can ship all the packages within the given number of days, while ensuring that the order of the packages on the conveyor belt is maintained.

Approach 1: Using Binary Search Algorithm to find the least weight allocated to each day.

The isPossibleSolution function checks if a given allocation of weight is a possible solution. It takes three parameters: the vector weights containing the weights of the packages, the number of days available for shipping, and the mid value representing the weight capacity being checked.

Inside the isPossibleSolution function:

It initializes daysCount and weightCount variables to keep track of the number of days used and the weight allocated on the current day, respectively.

It iterates through the weights vector using a for loop.

For each weight:

It checks if adding the weight to the current allocation weightCount exceeds the capacity mid. If not, it adds the weight to weightCount.

If adding the weight exceeds the capacity mid, it increments the daysCount and checks if the number of days exceeds the given limit or if the weight itself is greater than the maximum allowed mid. If either condition is true, it returns false.

If the allocation is valid, it starts a new allocation for the next day by updating weightCount with the current weight.

If all weights have been allocated within the given days, it returns true.

The shipWithinDays function calculates the minimum weight capacity required to ship all the packages within the given number of days. It takes two parameters: the vector weights containing the weights of the packages and the number of days available for shipping.

Inside the shipWithinDays function:

It initializes variables low, high, mid, ans, and weightSum.

It calculates the weightSum by iterating through the weights vector and summing all the weights.

It sets the upper bound high of the binary search as the weightSum.

It starts the binary search loop, which continues until low is less than or equal to high.

Inside the loop:

It calculates the mid value as the average of low and high.

It calls the isPossibleSolution function to check if the current mid value represents a possible solution.

If it is a valid solution, it updates the answer ans to the current mid value and searches for smaller weights by updating high to mid - 1.

If it is not a valid solution, it searches for larger weights by updating low to mid + 1.

After the binary search is completed, it returns the final answer ans, which represents the minimum weight capacity required to ship all the packages within the given number of days.

Overall, the code uses binary search to find the minimum weight capacity required to ship all the packages within the given number of days. The isPossibleSolution function checks if a given allocation of weight is a possible solution by simulating the shipping process.

The time complexity of the code is O(n log W), where 'n' is the number of weights and 'W' is the sum of all the weights.

The space complexity is O(1) as it uses a constant amount of additional space.