

1011. Capacity To Ship Packages Within D Days

A conveyor belt has packages that must be shipped from one port to another within `days` days.

The `weights[i]` package on the conveyor belt has a weight of `weights[i]`. Each day, we load the ship with packages on the conveyor belt (in the order given by `weights`). We may not load more weight than the maximum weight capacity of the ship.

Return the least weight capacity of the ship that will result in all the packages on the conveyor belt being shipped within `days` days.

Example 1:

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

Output: 15

Explanation: A ship capacity of 15 is the minimum to ship all the packages in 5 days like this:

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

2nd day: 6, 7

3rd day: 8

4th day: 9

5th day: 10

Note that the cargo must be shipped in the order given, so using a ship of capacity 14 and splitting the packages into parts like (2, 3, 4, 5), (1, 6, 7), (8), (9), (10) is not allowed.

Example 2:

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

Output: 6

Explanation: A ship capacity of 6 is the minimum to ship all the packages in 3 days like this:

1st day: 3, 2

2nd day: 2, 4

3rd day: 1, 4

Example 3:

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

Output: 3

Explanation:

```
1st day: 1
2nd day: 2
3rd day: 3
4th day: 1, 1
```

Constraints:

- $1 \leq \text{days} \leq \text{weights.length} \leq 5 * 10^4$
- $1 \leq \text{weights}[i] \leq 500$

```
class Solution:
    def shipWithinDays(self, weights: List[int], days: int) -> int:
        if len(weights)==days:
            return max(weights)
        lo = max(weights)
        hi = sum(weights)
        ans = -1
        while lo<=hi:
            mid = (lo+hi)//2

            if self.isValid(weights,mid,days):
                ans = mid
                hi = mid-1
            else:
                lo = mid+1
        return ans
    def isValid(self,weights,potentialWeight,days):
        tempSum = 0
        currDays = 1

        for i in range(len(weights)):
            tempSum = tempSum+weights[i]
            if tempSum>potentialWeight:
                currDays+=1
                tempSum = weights[i]

        if currDays>days:
            return False
        else:
            return True
```