

Problem 3647: Maximum Weight in Two Bags

Problem Information

Difficulty: Medium

Acceptance Rate: 59.60%

Paid Only: Yes

Tags: Array, Dynamic Programming

Problem Description

You are given an integer array `weights` and two integers `w1` and `w2` representing the **maximum** capacities of two bags.

Each item may be placed in **at most** one bag such that:

* Bag 1 holds **at most** `w1` total weight. * Bag 2 holds **at most** `w2` total weight.

Return the **maximum** total weight that can be packed into the two bags.

Example 1:

Input: weights = [1,4,3,2], w1 = 5, w2 = 4

Output: 9

Explanation:

* Bag 1: Place `weights[2] = 3` and `weights[3] = 2` as `3 + 2 = 5 <= w1` * Bag 2: Place `weights[1] = 4` as `4 <= w2` * Total weight: `5 + 4 = 9`

Example 2:

Input: weights = [3,6,4,8], w1 = 9, w2 = 7

Output: 15

****Explanation:****

* Bag 1: Place `weights[3] = 8` as `8 <= w1` * Bag 2: Place `weights[0] = 3` and `weights[2] = 4` as `3 + 4 = 7 <= w2` * Total weight: `8 + 7 = 15`

****Example 3:****

****Input:**** weights = [5,7], w1 = 2, w2 = 3

****Output:**** 0

****Explanation:****

No weight fits in either bag, thus the answer is 0.

****Constraints:****

* `1 <= weights.length <= 100` * `1 <= weights[i] <= 100` * `1 <= w1, w2 <= 300`

Code Snippets

C++:

```
class Solution {
public:
    int maxWeight(vector<int>& weights, int w1, int w2) {
        }
    };
}
```

Java:

```
class Solution {
public int maxWeight(int[] weights, int w1, int w2) {
        }
    };
}
```

Python3:

```
class Solution:  
    def maxWeight(self, weights: List[int], w1: int, w2: int) -> int:
```