

5612. Delivering Boxes from Storage to Ports

[My Submissions \(/contest/biweekly-contest-41/problems/delivering-boxes-from-storage-to-ports/submissions/\)](/contest/biweekly-contest-41/problems/delivering-boxes-from-storage-to-ports/submissions/)[Back to Contest \(/contest/biweekly-contest-41/\)](/contest/biweekly-contest-41/)

You have the task of delivering some boxes from storage to their ports using only one ship. However, this ship has a **limit** on the **number of boxes** and the **total weight** that it can carry.

You are given an array `boxes`, where `boxes[i] = [portsi, weighti]`, and three integers `portsCount`, `maxBoxes`, and `maxWeight`.

- `portsi` is the port where you need to deliver the i^{th} box and `weighti` is the weight of the i^{th} box.
- `portsCount` is the number of ports.
- `maxBoxes` and `maxWeight` are the respective box and weight limits of the ship.

The boxes need to be delivered **in the order they are given**. The ship will follow these steps:

- The ship will take some number of boxes from the `boxes` queue, not violating the `maxBoxes` and `maxWeight` constraints.
- For each loaded box **in order**, the ship will make a **trip** to the port the box needs to be delivered to and deliver it. If the ship is already at the correct port, no **trip** is needed, and the box can immediately be delivered.
- The ship then makes a return **trip** to storage to take more boxes from the queue.

The ship must end at storage after all the boxes have been delivered.

Return the **minimum** number of **trips** the ship needs to make to deliver all boxes to their respective ports.

Example 1:

Input: `boxes = [[1,1],[2,1],[1,1]]`, `portsCount = 2`, `maxBoxes = 3`, `maxWeight = 3`

Output: 4

Explanation: The optimal strategy is as follows:

– The ship takes all the boxes in the queue, goes to port 1, then port 2, then port 1 again, then returns to storage. 4 trips. So the total number of trips is 4.

Note that the first and third boxes cannot be delivered together because the boxes need to be delivered in order (i.e. the seco

Example 2:

Input: `boxes = [[1,2],[3,3],[3,1],[3,1],[2,4]]`, `portsCount = 3`, `maxBoxes = 3`, `maxWeight = 6`

Output: 6

Explanation: The optimal strategy is as follows:

– The ship takes the first box, goes to port 1, then returns to storage. 2 trips.
– The ship takes the second, third and fourth boxes, goes to port 3, then returns to storage. 2 trips.
– The ship takes the fifth box, goes to port 3, then returns to storage. 2 trips.
So the total number of trips is $2 + 2 + 2 = 6$.

Example 3:

Input: `boxes = [[1,4],[1,2],[2,1],[2,1],[3,2],[3,4]]`, `portsCount = 3`, `maxBoxes = 6`, `maxWeight = 7`

Output: 6

Explanation: The optimal strategy is as follows:

– The ship takes the first and second boxes, goes to port 1, then returns to storage. 2 trips.
– The ship takes the third and fourth boxes, goes to port 2, then returns to storage. 2 trips.
– The ship takes the fifth and sixth boxes, goes to port 3, then returns to storage. 2 trips.
So the total number of trips is $2 + 2 + 2 = 6$.

Example 4:

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Hard

Input: boxes = [[2,4],[2,5],[3,1],[3,2],[3,7],[3,1],[4,4],[1,3],[5,2]], portsCount = 5, maxBoxes = 5, maxWeight = 7

Output: 14

Explanation: The optimal strategy is as follows:

- The ship takes the first box, goes to port 2, then storage. 2 trips.
 - The ship takes the second box, goes to port 2, then storage. 2 trips.
 - The ship takes the third and fourth boxes, goes to port 3, then storage. 2 trips.
 - The ship takes the fifth box, goes to port 3, then storage. 2 trips.
 - The ship takes the sixth and seventh boxes, goes to port 3, then port 4, then storage. 3 trips.
 - The ship takes the eighth and ninth boxes, goes to port 1, then port 5, then storage. 3 trips.
- So the total number of trips is 2 + 2 + 2 + 2 + 3 + 3 = 14.

Constraints:

- $1 \leq \text{boxes.length} \leq 10^5$
- $1 \leq \text{portsCount}, \text{maxBoxes}, \text{maxWeight} \leq 10^5$
- $1 \leq \text{ports}_i \leq \text{portsCount}$
- $1 \leq \text{weights}_i \leq \text{maxWeight}$

JavaScript



```
1  /**
2   * @param {number[][]} boxes
3   * @param {number} portsCount
4   * @param {number} maxBoxes
5   * @param {number} maxWeight
6   * @return {number}
7   */
8  var boxDelivering = function(boxes, portsCount, maxBoxes, maxWeight) {
9
10 };
```

☐ Custom Testcase

Use Example Testcases

Run

Submit