(/) Explore Problems(/problemset/all/) Mock(/interview/) Contest











5641. Maximum Units on a Truck

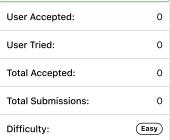
 $[numberOfBoxes_i, numberOfUnitsPerBox_i]$:

My Submissions (/contest/weekly-contest-222/problems/maximum-units-on-a-truck/submissions/) Back to Contest (/contest/weekly-contest-222/) You are assigned to put some amount of boxes onto one truck. You are given a 2D array boxTypes , where boxTypes [i] =

- $number0fBoxes_i$ is the number of boxes of type i.
- $numberOfUnitsPerBox_i$ is the number of units in each box of the type i.

You are also given an integer truckSize, which is the maximum number of boxes that can be put on the truck. You can choose any boxes to put on the truck as long as the number of boxes does not exceed truckSize.

Return the maximum total number of units that can be put on the truck.



Example 1:

```
Input: boxTypes = [[1,3],[2,2],[3,1]], truckSize = 4
Output: 8
Explanation: There are:
- 1 box of the first type that contains 3 units.
- 2 boxes of the second type that contain 2 units each.
- 3 boxes of the third type that contain 1 unit each.
You can take all the boxes of the first and second types, and one box of the third type.
The total number of units will be = (1 * 3) + (2 * 2) + (1 * 1) = 8.
```

Example 2:

```
Input: boxTypes = [[5,10],[2,5],[4,7],[3,9]], truckSize = 10
Output: 91
```

Constraints:

- 1 <= boxTypes.length <= 1000
- 1 <= number0fBoxes_i, number0fUnitsPerBox_i <= 1000
- 1 <= truckSize <= 10⁶

```
JavaScript
                                                                                                                          4
                                                                                                                                \mathfrak{C}
1 • /**
     * @param {number[][]} boxTypes
2
 3
     * @param {number} truckSize
     * @return {number}
 4
 5
 6 •
    const maximumUnits = (boxTypes, truckSize) => {
7
        let n = boxTypes.length;
 8
        boxTypes.sort((a, b) \Rightarrow b[1] - a[1]);
9
        let res = curSize = 0;
10
        let lastIdx = n - 1;
11 ▼
        for (let i = 0; i < n; i++) {
12
             if (curSize >= truckSize) {
                 lastIdx = i - 1;
13
14
                 break:
15
16
             let num = boxTypes[i][0];
             let units = boxTypes[i][1];
17
18
             curSize += num;
             res += units * num;
19
20
21
        if (curSize == truckSize) return res;
        if (curSize > truckSize) {
22 1
23
             let units2 = boxTypes[lastIdx][1];
             res -= units2 * (curSize - truckSize);
24
25
26
        return res;
27
    };
```