

Leetcode 1049. Last Stone Weight II

Success Details >

Runtime: 1 ms, faster than 98.42% of Java online submissions for Last Stone Weight II.

Memory Usage: 36.2 MB, less than 83.66% of Java online submissions for Last Stone Weight II.

Next challenges:

- Knight Probability in Chessboard
- Distinct Subsequences II
- Maximum Subarray Sum with One Deletion

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
12/05/2020 23:03	Accepted	1 ms	36.2 MB	java
12/05/2020 23:02	Accepted	2 ms	38.7 MB	java
12/05/2020 23:01	Accepted	3 ms	38.4 MB	java
12/05/2020 23:00	Accepted	1 ms	36.2 MB	java

```
1 class Solution {
2
3     // 0-1 knapsack
4     // Divide into two sets, positive and negative
5     public int lastStoneWeightII(int[] stones) {
6         int sum = 0;
7         for (int stone: stones)
8             sum += stone;
9
10        boolean[] dp = new boolean[sum + 1];
11        dp[0] = true;
12        int tmp = 0;
13        for (int stone: stones) {
14            tmp += stone;
15            for (int i = tmp; i >= 0; i--) {
16                if (i - stone >= 0)
17                    dp[i] |= dp[i - stone];
18            }
19        }
20    }
```

Your previous code was restored from your local storage. [Reset to default](#)

Testcase: Run Code Result Debugger

Accepted Runtime: 0 ms

Your input: [2,7,4,1,8,1]

Output: 1

Expected: 1

Diff

Console How to create a testcase

Run Code Submit

DP, 0-1 knapsack problem

Divide the numbers into two groups, positive and negative (multiple -1), and then find the **smallest possible** result of $\text{Sum}(\text{positive}) + \text{Sum}(\text{negative})$, call this result S.

We have:

$$S = \text{Sum}(\text{positive}) + \text{Sum}(\text{negative})$$

$$\text{Sum}(\text{num}) = \text{Sum}(\text{positive}) - \text{Sum}(\text{negative})$$

$$\text{Then } S = 2 * \text{Sum}(\text{positive}) - \text{Sum}(\text{num})$$

Let $\text{dp}[0 \dots \text{Sum}(\text{num})]$, $\text{dp}[i]$ denotes whether i can be calculated

$$\text{dp}[i] = \text{dp}[i] \mid \mid \text{dp}[i - \text{stone}] \text{ for all stones}$$

Finally, we only need to find the largest possible i ($i \leq \text{sum}/2$) and return $\text{sum} - i * 2$.