5856. Minimum Number of Work Sessions to Finish the Tasks

My Submissions (/contest/weekly-contest-256/problems/minimum-number-of-work-sessions-to-finish-the-tasks/submissions/)

Back to Contest (/contest/weekly-contest-256/)

There are `n` tasks assigned to you. The task times are represented as an integer array `tasks` of length `n`, where the i^{th} task takes `tasks[i]` hours to finish. A **work session** is when you work for **at most** `sessionTime` consecutive hours and then take a break.

You should finish the given tasks in a way that satisfies the following conditions:

- If you start a task in a work session, you must complete it in the **same** work session.
- You can start a new task **immediately** after finishing the previous one.
- You may complete the tasks in **any order**.

Given `tasks` and `sessionTime`, return *the **minimum** number of **work sessions** needed to finish all the tasks following the conditions above.*

The tests are generated such that `sessionTime` is **greater** than or **equal** to the **maximum** element in `tasks[i]`.

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

Example 1:

```
Input: tasks = [1,2,3], sessionTime = 3
Output: 2
Explanation: You can finish the tasks in two work sessions.
- First work session: finish the first and the second tasks in 1 + 2 = 3 hours.
- Second work session: finish the third task in 3 hours.
```

Example 2:

```
Input: tasks = [3,1,3,1,1], sessionTime = 8
Output: 2
Explanation: You can finish the tasks in two work sessions.
- First work session: finish all the tasks except the last one in 3 + 1 + 3 + 1 = 8 hours.
- Second work session: finish the last task in 1 hour.
```

Example 3:

```
Input: tasks = [1,2,3,4,5], sessionTime = 15
Output: 1
Explanation: You can finish all the tasks in one work session.
```

Constraints:

- `n == tasks.length`
- `1 <= n <= 14`
- `1 <= tasks[i] <= 10`
- `max(tasks[i]) <= sessionTime <= 15`

JavaScript

```
const minSessions = (tasks, sessionTime) => {
    let n = tasks.length;
    let dp = Array(1 << n).fill(Number.MAX_SAFE_INTEGER);
    let sum = Array(1 << n).fill(0);
```

```
    for (let i = 0; i < 1 << n; i++) {
        for (let j = 0; j < n; j++) {
            if (1 & (i >> j)) sum[i] += tasks[j];
        }
    }
    dp[0] = 0;
    for (let i = 0; i < 1 << n; i++) {
        for (let j = i; j > 0; j = (j - 1) & i) {
            let valid = sum[j] <= sessionTime;
            if (valid) {
                dp[i] = Math.min(dp[i], dp[i - j] + 1);
            }
        }
    }
    return dp[(1 << n) - 1];
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/545963141/) | More Details > (/submissions/detail/545963141/)

Share your acceptance!