Problem 1723: Find Minimum Time to Finish All Jobs

Problem Information

Difficulty: Hard
Acceptance Rate: 44.71%
Paid Only: No
Tags: Array, Dynamic Programming, Backtracking, Bit Manipulation, Bitmask

Problem Description

You are given an integer array `jobs`, where `jobs[i]` is the amount of time it takes to complete the `ith` job.

There are `k` workers that you can assign jobs to. Each job should be assigned to **exactly** one worker. The **working time** of a worker is the sum of the time it takes to complete all jobs assigned to them. Your goal is to devise an optimal assignment such that the **maximum working time** of any worker is **minimized**.

_Return the**minimum** possible **maximum working time** of any assignment. _

Example 1:

Input: jobs = [3,2,3], k = 3 **Output:** 3 **Explanation:** By assigning each person one job, the maximum time is 3.

Example 2:

Input: jobs = [1,2,4,7,8], k = 2 **Output:** 11 **Explanation:** Assign the jobs the following way: Worker 1: 1, 2, 8 (working time = 1 + 2 + 8 = 11) Worker 2: 4, 7 (working time = 4 + 7 = 11) The maximum working time is 11.

Constraints:

* `1 <= k <= jobs.length <= 12` * `1 <= jobs[i] <= 107`

Code Snippets

C++:

```
class Solution {
public:
int minimumTimeRequired(vector<int>& jobs, int k) {

}
};
```

Java:

```
class Solution {
public int minimumTimeRequired(int[] jobs, int k) {

}
}
```

Python3:

```
class Solution:
def minimumTimeRequired(self, jobs: List[int], k: int) -> int:
```