3.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

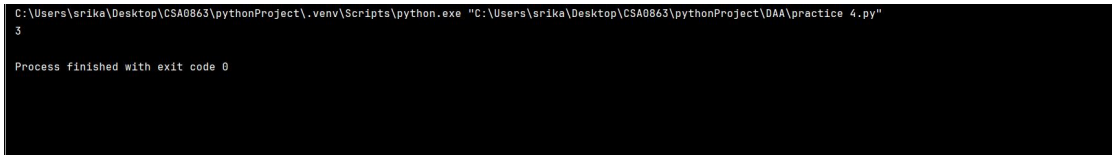
Output: 3

Program:

```
def can_schedule(jobs, k, max_time):
    workers = [0] * k
    def backtrack(index):
        if index == len(jobs):
            return True
        for i in range(k):
            if workers[i] + jobs[index] <= max_time:
                workers[i] += jobs[index]
                if backtrack(index + 1):
                    return True
                workers[i] -= jobs[index]
            if workers[i] == 0:
                break
        return False
    return backtrack(0)
def minimumTimeRequired(jobs, k):
    left, right = max(jobs), sum(jobs)
    while left < right:
```

```
        mid = (left + right) // 2
        if can_schedule(jobs, k, mid):
            right = mid
        else:
            left = mid + 1
    return left
jobs = [3, 2, 3]
k = 3
print(minimumTimeRequired(jobs, k))
```

Output:



Time complexity:

O(log(sum(jobs))-max(jobs).k^n)