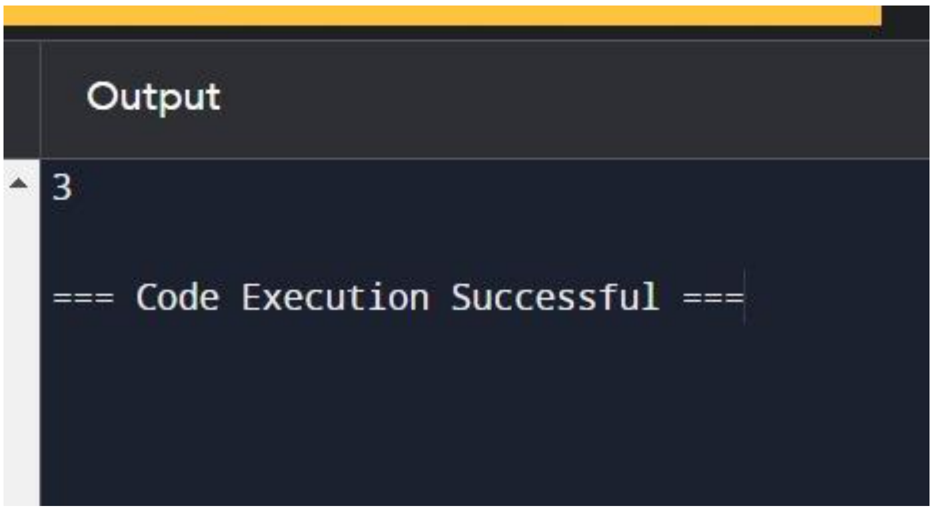
196. You are given an integer array jobs, where jobs[i] is the amount of time it takes to complete the ith job. There are k workers that you can assign jobs to. Each job should be assigned to exactly one worker. The working time of a worker is the sum of the time it takes to complete all jobs assigned 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.

Program:

```
def min_max_working_time(jobs, k):

    def is_valid(limit):

        workers = [0] * k

        if backtrack(0, jobs, workers, limit):

            return True

        return False

    def backtrack(idx, jobs, workers, limit):

        if idx == len(jobs):

            return True

 job = jobs[idx]

        for i in range(k):

            if workers[i] + job <= limit:

                workers[i] += job

                if backtrack(idx + 1, jobs, workers, limit):

                    return True

                workers[i] -= job

                if workers[i] == 0:

                    break

        return False

    jobs.sort(reverse=True)

    left, right = max(jobs), sum(jobs)

    while left < right:
        mid = left + (right - left) // 2
        if is_valid(mid):
            right = mid
        else:
            left = mid + 1
    return left
# Example
jobs1 = [3, 2, 3]
k1 = 3
print(min_max_working_time(jobs1, k1))
```

Output:



Output

```
3

=== Code Execution Successful ===
```

Time complexity: O(nlog(total job time)),