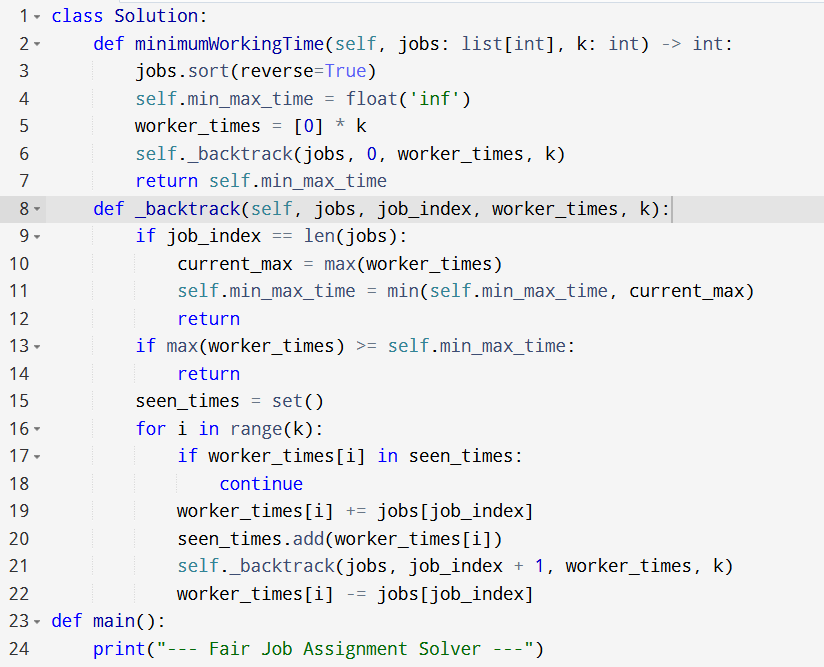
**5.3 Fair Job Assignment**

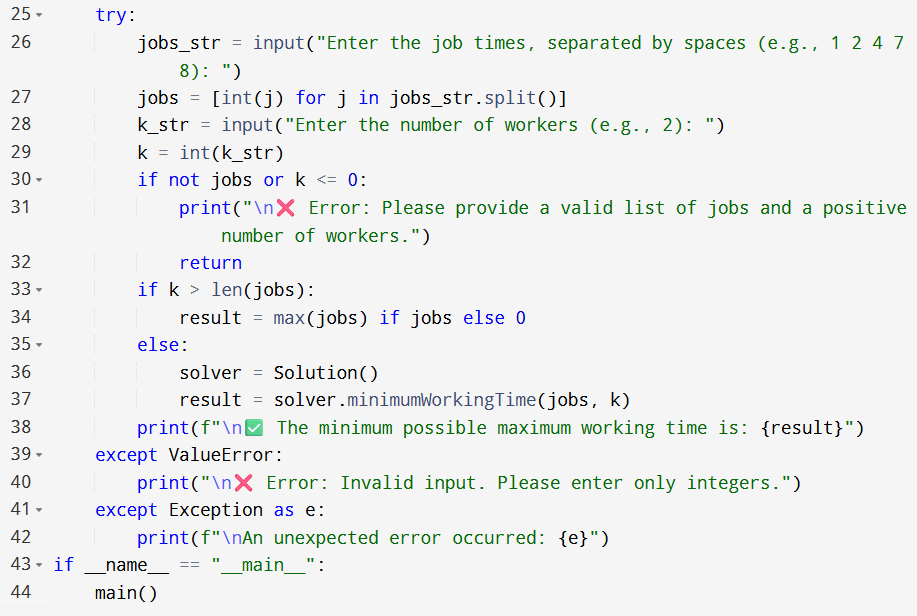
**Aim:** The aim is to assign jobs to k workers to minimize the maximum total working time of any single worker.

**Algorithm:**

1. Sort the jobs in non-increasing order (largest to smallest).  
   This helps assign heavier jobs earlier and prune faster.
2. Initialize an array workloads = [0, 0, …, 0] of size k,  
   where workloads[w] represents the total time assigned to worker w.
3. Define a recursive function dfs(i) meaning:
   * Assign job i (with time jobs[i]) to some worker w.
   * Update workloads[w] += jobs[i].
   * Recurse to assign i+1.
   * Backtrack afterwards.
4. At each step, track the current maximum workload across all workers.
   * If this already exceeds the best answer found so far, prune (stop exploring this branch).
5. Continue until all jobs are assigned.  
   The minimum possible value of the maximum workload is the answer.

**Program:**

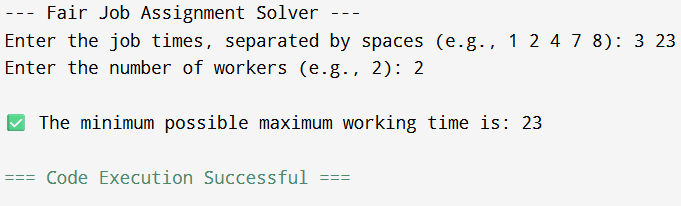
****



**Input:**

* Enter the job times, separated by spaces (e.g., 1 2 4 7 8):

**Output:**

****

**Result:** Thus, the program is executed successfully and output is verified.

**Performance analysis:**

* Time Complexity: O(k^n)
* Space Complexity: O(n+k).