

Task Name

Balanced Segment Partition 1

Difficulty

Hard

Problem Description

Split the array into exactly k contiguous parts so the largest part sum is as small as possible.

Problem Statement

You are given an array a of n non-negative integers and an integer k .

Split the array into exactly k non-empty contiguous segments.

Let the value of a split be the largest segment sum among those k segments. Your task is to find the minimum possible value of this largest segment sum.

Input Format

- The first line contains two integers n and k .
- The second line contains n space-separated integers $a[0]$, $a[1]$, ..., $a[n - 1]$.

Constraints

- $1 \leq n \leq 1000$
- $0 \leq a[i] \leq 10^6$
- $1 \leq k \leq \min(50, n)$

Output Format

Print one integer: the minimum possible largest segment sum after splitting the array into exactly k contiguous non-empty segments.

Example 1

Input

6 3 4 2 7 3 6 1

Output

10

Explanation

One optimal split is: - [4, 2] with sum 6 - [7] with sum 7 - [3, 6, 1] with sum 10

The largest segment sum is 10, and no valid split can do better.

Example 2

Input

5 2 1 4 4 7 2

Output

9

Explanation

An optimal split is: - [1, 4, 4] with sum 9 - [7, 2] with sum 9

So the minimum possible largest segment sum is 9.

Tags

Arrays, Binary Search, Greedy

Reference Solution Idea

Binary search the answer.

- The smallest possible answer is $\max(\mathbf{a})$ because every segment must contain at least one element.
- The largest possible answer is $\text{sum}(\mathbf{a})$ by taking the whole array as one segment before refining.
- For a candidate value mid , greedily build segments from left to right.
- If adding the next element would make the segment sum exceed mid , start a new segment.
- If you can finish using at most k segments, then mid is feasible.

This monotonic property makes binary search work.

Complexity

- Time: $O(n \log(\text{sum}(\mathbf{a})))$
- Space: $O(1)$ extra space

Suggested testcase roles

- input00/output00 -> Sample, strength 0
- input01/output01 -> Sample, strength 0
- input02/output02 -> Hidden, strength 50
- input03/output03 -> Hidden, strength 50