

## 410. Split Array Largest Sum

Given an integer array `nums` and an integer `k`, split `nums` into `k` non-empty subarrays such that the largest sum of any subarray is **minimized**.

Return *the minimized largest sum of the split*.

A **subarray** is a contiguous part of the array.

### Example 1:

**Input:** `nums = [7,2,5,10,8]`, `k = 2`

**Output:** 18

**Explanation:** There are four ways to split `nums` into two subarrays.

The best way is to split it into `[7,2,5]` and `[10,8]`, where the largest sum among the two subarrays is only 18.

### Example 2:

**Input:** `nums = [1,2,3,4,5]`, `k = 2`

**Output:** 9

**Explanation:** There are four ways to split `nums` into two subarrays.

The best way is to split it into `[1,2,3]` and `[4,5]`, where the largest sum among the two subarrays is only 9.

### Constraints:

- $1 \leq \text{nums.length} \leq 1000$
- $0 \leq \text{nums}[i] \leq 10^6$
- $1 \leq k \leq \min(50, \text{nums.length})$