Given an array nums which consists of non-negative integers and an integer m, you can split the array into m non-empty continuous subarrays.

Write an algorithm to minimize the largest sum among these m subarrays.

**Example 1:**

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

**Output:** 9

**Example 3:**

**Input:** nums = [1,4,4], m = 3

**Output:** 4

**Constraints:**

* 1 <= nums.length <= 1000
* 0 <= nums[i] <= 106
* 1 <= m <= min(50, nums.length)