Problem 410: Split Array Largest Sum

Problem Information

Difficulty: Hard
Acceptance Rate: 59.27%
Paid Only: No
Tags: Array, Binary Search, Dynamic Programming, Greedy, Prefix Sum

Problem Description

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

Code Snippets

C++:

```
class Solution {
public:
int splitArray(vector<int>& nums, int k) {

}
};
```

Java:

```
class Solution {
public int splitArray(int[] nums, int k) {

}
}
```

Python3:

```
class Solution:
def splitArray(self, nums: List[int], k: int) -> int:
```