Problem 1231: Divide Chocolate

Problem Information

Difficulty: Hard
Acceptance Rate: 60.34%
Paid Only: Yes
Tags: Array, Binary Search

Problem Description

You have one chocolate bar that consists of some chunks. Each chunk has its own sweetness given by the array `sweetness`.

You want to share the chocolate with your `k` friends so you start cutting the chocolate bar into `k + 1` pieces using `k` cuts, each piece consists of some **consecutive** chunks.

Being generous, you will eat the piece with the **minimum total sweetness** and give the other pieces to your friends.

Find the **maximum total sweetness** of the piece you can get by cutting the chocolate bar optimally.

Example 1:

Input: sweetness = [1,2,3,4,5,6,7,8,9], k = 5 **Output:** 6 **Explanation:** You can divide the chocolate to [1,2,3], [4,5], [6], [7], [8], [9]

Example 2:

Input: sweetness = [5,6,7,8,9,1,2,3,4], k = 8 **Output:** 1 **Explanation:** There is only one way to cut the bar into 9 pieces.

Example 3:

Input: sweetness = [1,2,2,1,2,2,1,2,2], k = 2 **Output:** 5 **Explanation:** You can divide the chocolate to [1,2,2], [1,2,2], [1,2,2]

Constraints:

* `0 <= k < sweetness.length <= 104` * `1 <= sweetness[i] <= 105`

Code Snippets

C++:

```
class Solution {
public:
int maximizeSweetness(vector<int>& sweetness, int k) {

}
};
```

Java:

```
class Solution {
public int maximizeSweetness(int[] sweetness, int k) {

}
}
```

Python3:

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