Problem 3366: Minimum Array Sum

Problem Information

Difficulty: Medium
Acceptance Rate: 30.41%
Paid Only: No
Tags: Array, Dynamic Programming

Problem Description

You are given an integer array `nums` and three integers `k`, `op1`, and `op2`.

You can perform the following operations on `nums`:

* **Operation 1** : Choose an index `i` and divide `nums[i]` by 2, **rounding up** to the nearest whole number. You can perform this operation at most `op1` times, and not more than **once** per index. * **Operation 2** : Choose an index `i` and subtract `k` from `nums[i]`, but only if `nums[i]` is greater than or equal to `k`. You can perform this operation at most `op2` times, and not more than **once** per index.

Note: Both operations can be applied to the same index, but at most once each.

Return the **minimum** possible **sum** of all elements in `nums` after performing any number of operations.

Example 1:

Input: nums = [2,8,3,19,3], k = 3, op1 = 1, op2 = 1

Output: 23

Explanation:

* Apply Operation 2 to `nums[1] = 8`, making `nums[1] = 5`. * Apply Operation 1 to `nums[3] = 19`, making `nums[3] = 10`. * The resulting array becomes `[2, 5, 3, 10, 3]`, which has the minimum possible sum of 23 after applying the operations.

Example 2:

Input: nums = [2,4,3], k = 3, op1 = 2, op2 = 1

Output: 3

Explanation:

* Apply Operation 1 to `nums[0] = 2`, making `nums[0] = 1`. * Apply Operation 1 to `nums[1] = 4`, making `nums[1] = 2`. * Apply Operation 2 to `nums[2] = 3`, making `nums[2] = 0`. * The resulting array becomes `[1, 2, 0]`, which has the minimum possible sum of 3 after applying the operations.

Constraints:

* `1 <= nums.length <= 100` * `0 <= nums[i] <= 105` * `0 <= k <= 105` * `0 <= op1, op2 <= nums.length`

Code Snippets

C++:

```
class Solution {
public:
int minArraySum(vector<int>& nums, int k, int op1, int op2) {

}
};
```

Java:

```
class Solution {
public int minArraySum(int[] nums, int k, int op1, int op2) {

}
}
```

Python3:

```
class Solution:
def minArraySum(self, nums: List[int], k: int, op1: int, op2: int) -> int:
```