Minimize Coin Removal for Bounded Pile Differences

18 June 2025 09:03 AM

Given an array **arr[]** of integers, where each element represents the number of coins in a pile. You are also given an integer **k.**

Your task is to remove the minimum number of **coins** such that the absolute difference between the number of coins in any two remaining piles is at most **k.**

Note: You can also remove a pile by removing all the coins of that pile.

Examples:

Input: *arr[] = [2, 2, 2, 2], k= 0*

Output: *0*

Explanation: *For all piles the difference in the number of coins is = 0. So, no need to remove any coins.*

Input: *arr[] = [1, 5, 1, 2, 5, 1], k = 3*

Output: *2*

Explanation: *If we remove one coin each from both the piles containing 5 coins, then for any two piles the absolute difference in the number of coins is <= 3.*

Try it on GfG Practice

[Approach]: Binary Search and Prefix Sum

We first sort the array to handle piles in order, then for each pile, we assume it to be the smallest remaining pile and compute the cost to remove all smaller piles entirely and trim larger ones to ensure no pile exceeds $arr[i] + k$. *For each index, we efficiently calculate the coins to remove and select the minimum among them.*

Output

2

Time Complexity: O(nlog(n)) due to sorting and performing binary search for each of the n elements.

Space complexity: O(n) due to the prefix sum array used to store cumulative sums of the sorted array.

SDE Sheet - Coin Piles

From <https://www.geeksforgeeks.org/dsa/remove-minimum-coins-such-that-absolute-difference-between-any-two-piles-is-less-than-k/>

Coin Piles

Difficulty: **Medium** Accuracy: **20.51%** Submissions: **44K+** Points: **4** Average Time: **20m**

You are given an array **arr[]** of integers, where each element represents the number of coins in a pil
You are also given an integer **k.**
Your task is to remove the minimum number of **coins** such that the absolute difference between the number of coins in any two updated piles is at most **k.**

Note: You can also remove a pile by removing all the coins of that pile.

Examples:

Input: arr[] = [2, 2, 2, 2], k = 0
Output: 0
Explanation: For any two piles the difference in the number of coins is <= 0. So no need to remove any coin.

Input: arr[] = [1, 5, 1, 2, 5, 1], k = 3
Output: 2
Explanation: If we remove one coin each from both the piles containing 5 coins, then for any two piles the absolute difference in the number of coins is <= 3.

Constraints:

$1 \leq arr.size() \leq 10^5$

$1 \leq arr[i] \leq 10^4$

$0 \leq k \leq 10^4$

Try more examples

Expected Complexities

Company Tags

Menu

Topic Tags

Related Articles

Python3 Start Timer

```python
import bisect

class Solution:
    def minimumCoins(self, arr, k):
        arr.sort()
        n = len(arr)
        prefix = [0] * n
        prefix[0] = arr[0]

        for i in range(1, n):
            prefix[i] = prefix[i - 1] + arr[i]

        ans = float('inf')
        prev = 0

        for i in range(n):
            if i > 0 and arr[i] == arr[i - 1]:
                continue
            if i > 0:
                prev = prefix[i - 1]

            pos = bisect.bisect_right(arr, arr[i] + k, i, n)
            totalToRemove = prev
            if pos < n:
                totalToRemove += prefix[n - 1] - prefix[pos - 1] - (n - pos) * (arr[i] + k
            ans = min(ans, totalToRemove)

        return ans
```

Custom Input Compile & Run Submit

Report An Issue

If you are facing any issue on this page. Please let us know.

Menu