Description Submission view

Due date: Cuma, 17 Ekim 2025, 4:20 PM
Requested files: G2W2Q1.java (Download)
Type of work: Individual work

Neo-Metropolis

Story

Neo-Metropolis' financial analysts track daily cash-flow "activity levels" as an integer sequence. To study stability, they focus on time windows whose total activity falls within a target band. Scanning every window naïvely would be far too slow for city-scale data, so the analysts need an algorithmic ritual—swift and exact—to count all qualifying windows.

Task (Problem Definition)

You are given an integer array `nums` and two integers `L` (lower bound) and `R` (upper bound). Count how many contiguous subarrays have a sum in the closed interval `[L, R]`.

Solution Requirement

Your algorithm must run in `O(N log N)` time. Quadratic approaches (e.g., checking all subarrays) are not acceptable.

Required Output

A single integer: the number of subarrays whose sum is in `[L, R]`.

Example Scene

nums = `[-2, 5, -1]`
L = `-2`, **R** = `2`
Expected Count: `3`.

Explanation (sums within [−2, 2]):

- `nums[0] = -2` → sum `-2`
- `nums[2] = -1` → sum `-1`
- `nums[0] + nums[1] + nums[2] = -2 + 5 + (-1) = 2`

Other subarrays (outside the band):

- `nums[1] = 5` → `5`
- `nums[0] + nums[1] = 3`
- `nums[1] + nums[2] = 4`

Final Answer: `3`.

Requested files

G2W2Q1.java

```java
import java.util.*;

public class G2W2Q1 {

    public static long countRangeSum(int[] nums, long L, long R) {
        if (nums == null) throw new IllegalArgumentException("nums must not be null");
        int n = nums.length;

        long[] sums = new long[n + 1];
        for (int i = 0; i < n; i++) sums[i + 1] = sums[i] + nums[i];

        long[] tmp = new long[sums.length];
        return sortAndCount(sums, 0, sums.length, L, R, tmp);
    }

    private static long sortAndCount(long[] sums, int lo, int hi, long L, long R, long[] tmp) {

    }

    private static void runScenario(String title, int[] nums, int L, int R, long expectedOrNeg) {
        long ans = countRangeSum(nums, L, R);
        System.out.println("=== " + title + " ===");
        System.out.println("nums = " + Arrays.toString(nums) + " | L=" + L + " | R=" + R);
        System.out.println("Count: " + ans);
        if (expectedOrNeg >= 0) {
            System.out.println("Matches expected? " + (ans == expectedOrNeg) + "  Expected: " + expectedOrNeg);
        }
        System.out.println();
    }

    public static void main(String[] args) {
        runScenario("Scenario 1 (PDF)", new int[]{-2, 5, -1}, -2, 2, 3);
        runScenario("Single in-range", new int[]{0}, 0, 0, 1);
        runScenario("Single out", new int[]{5}, -2, 2, 0);
    }
}

```

VPL