Description Submission view

Available from: Cuma, 24 Ekim 2025, 4:15 PM
Due date: Cuma, 24 Ekim 2025, 6:00 PM
Requested files: G2W3Q1.java (Download)
Type of work: Individual work

Top K Frequent Elements

Story

A data scientist at "Trendify," a leading social media analytics firm, is monitoring a real-time stream of user posts. To identify emerging trends, their system needs to process a massive array of keywords (represented as integers in `nums`) generated every minute. Their critical task is to find the `k` most frequent keywords from this list to display on the company's "Trending Now" dashboard. The list is huge, so the solution must be extremely fast.

Task (Problem Definition)

Given an integer array `nums` and an integer `k`, return the `k` most frequent elements. You may return the answer in any order.

Required Output

- A list of integers representing the `k` most frequent elements.

Example Scenarios

Scenario 1
Input: `nums = [1,1,1,2,2,3]`, `k = 2`
Expected Output: `[1, 2]`
Explanation (Frequency Count):

- `1` appears 3 times.
- `2` appears 2 times.
- `3` appears 1 time.
  The 2 most frequent elements are `1` and `2`.

Scenario 2
Input: `nums = [1]`, `k = 1`
Expected Output: `[1]`
Explanation: `1` is the only element, so it is the 1st most frequent.

Scenario 3
Input: `nums = [1,2,1,2,1,2,3,1,3,2]`, `k = 2`
Expected Output: `[1, 2]`
Explanation (Frequency Count):

- `1` appears 4 times.
- `2` appears 4 times.
- `3` appears 2 times.
  The 2 most frequent elements are `1` and `2`.

Requested files

G2W3Q1.java

?

```java
import java.util.*;

public class G2W3Q1 {

    public static int[] topKFrequent(int[] nums, int k) {

    }

    private static void runScenario(String title, int[] nums, int k, int[] expectedOrNull) {
        int[] ans = topKFrequent(nums, k);
        System.out.println("=== " + title + " ===");
        System.out.println("nums = " + Arrays.toString(nums) + ", k = " + k);
        System.out.println("TopK = " + Arrays.toString(ans));
        if (expectedOrNull != null) {
            boolean ok = equalsIgnoreOrder(ans, expectedOrNull);
            System.out.println("Matches expected (order-free)? " + ok
                    + "  Expected: " + Arrays.toString(expectedOrNull));
        }
        System.out.println();
    }

    private static boolean equalsIgnoreOrder(int[] a, int[] b) {
        if (a == null || b == null || a.length != b.length) return false;
        int[] aa = a.clone(), bb = b.clone();
        Arrays.sort(aa); Arrays.sort(bb);
        return Arrays.equals(aa, bb);
    }

    public static void main(String[] args) {
        runScenario("Scenario 1 (PDF)", new int[]{1,1,1,2,2,3}, 2, new int[]{1,2});
        runScenario("Scenario 2 (PDF)", new int[]{1}, 1, new int[]{1});
        runScenario("Scenario 3 (PDF)", new int[]{1,2,1,2,1,2,3,1,3,2}, 2, new int[]{1,2});
        runScenario("Ties > k", new int[]{1,2,1,2,1,2,3,4}, 2, new int[]{1,2});
        runScenario("k > unique", new int[]{5,5,6}, 5, new int[]{5,6});
        runScenario("Negatives & zeros", new int[]{0,-1,-1,-2,-2,-2}, 2, new int[]{-2,-1});
        runScenario("Random-ish", new int[]{7,7,7,8,8,9,10,10,10,10}, 3, new int[]{10,7,8});
    }
}
```

VPL