Frequency Tracker (/contest/weekly-contest-344/problems/frequency-tracker/)

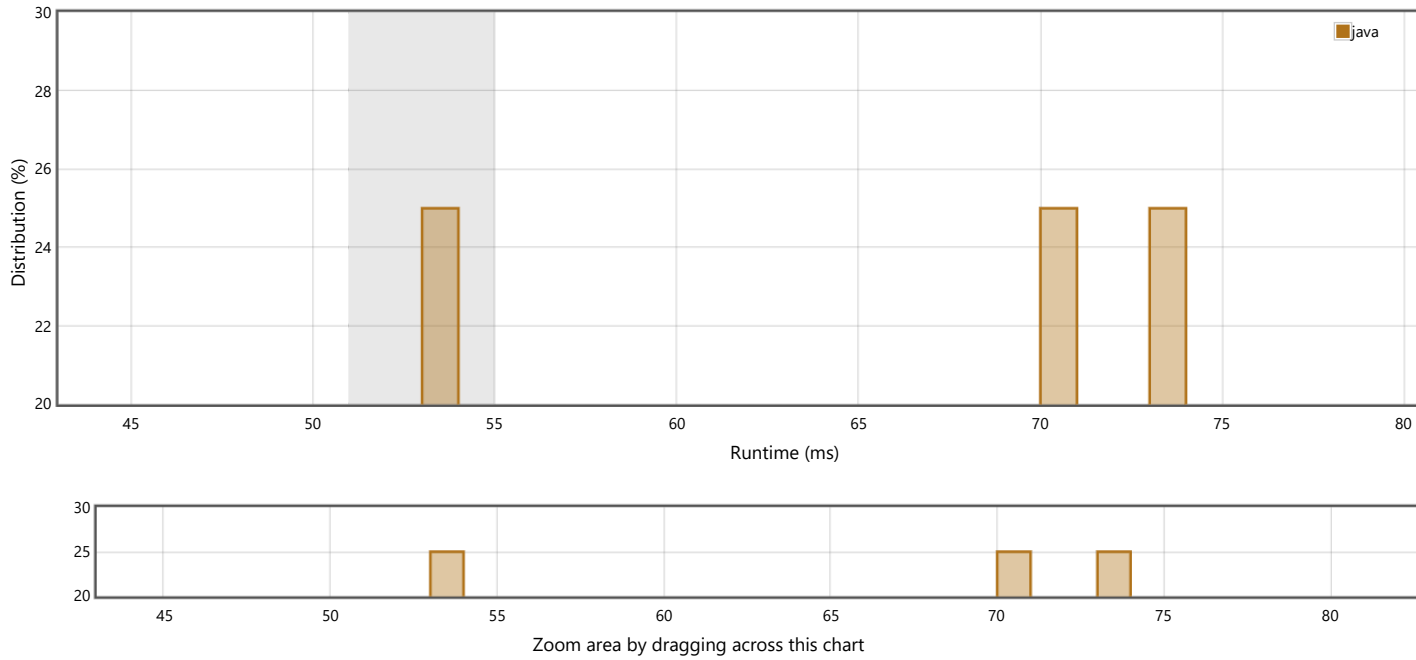## Submission Detail

1117 / 1117 test cases passed.

Runtime: **53 ms**
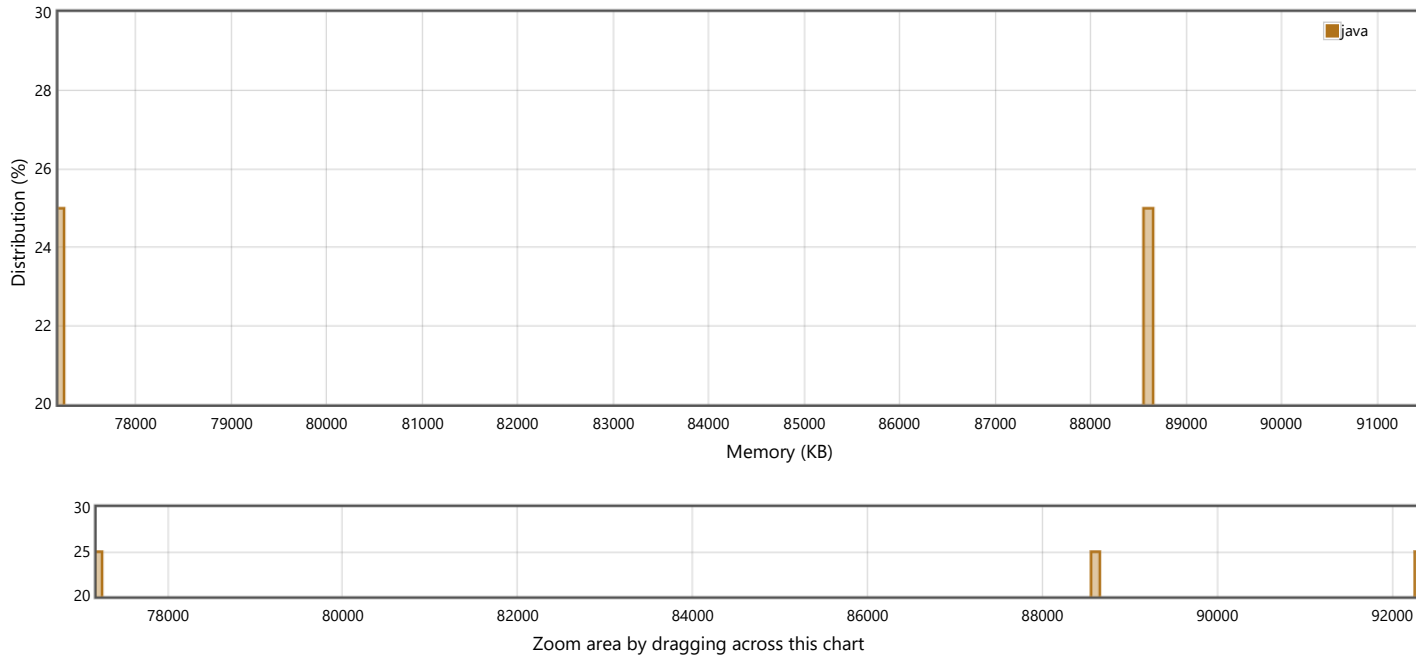Memory Usage: **108.4 MB**

Status: **Accepted**

Submitted: **1 day ago**

Accepted Solutions Runtime Distribution

Accepted Solutions Memory Distribution

Invite friends to challenge **Frequency Tracker**

◁ **5**

Submitted Code: 1 day ago

Language: java

Edit Code

```
1    class FrequencyTracker {
```

```java
    private Map<Integer, Integer> numToFreqMap; // map of number to frequency
    private Map<Integer, Integer> freqToCountMap; // map of frequency to count

    public FrequencyTracker() {
        numToFreqMap = new HashMap<>();
        freqToCountMap = new HashMap<>();
    }

    public void add(int number) {
        int freq = numToFreqMap.getOrDefault(number, 0);
        int newFreq = freq + 1;
        numToFreqMap.put(number, newFreq);
        freqToCountMap.put(newFreq, freqToCountMap.getOrDefault(newFreq, 0) + 1);
        if (freq > 0) {
            freqToCountMap.put(freq, freqToCountMap.get(freq) - 1);
            if (freqToCountMap.get(freq) == 0) {
                freqToCountMap.remove(freq);
            }
        }
    }

    public void deleteOne(int number) {
        int freq = numToFreqMap.getOrDefault(number, 0);
        if (freq > 0) {
            numToFreqMap.put(number, freq - 1);
            freqToCountMap.put(freq, freqToCountMap.get(freq) - 1);
            if (freqToCountMap.get(freq) == 0) {
                freqToCountMap.remove(freq);
            }
            if (freq > 1) {
                freqToCountMap.put(freq - 1, freqToCountMap.getOrDefault(freq - 1, 0) + 1);
            }
        }
    }

    public boolean hasFrequency(int frequency) {
        return freqToCountMap.containsKey(frequency);
    }
}
```