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

## **Submission Detail**

1117 / 1117 test cases passed.

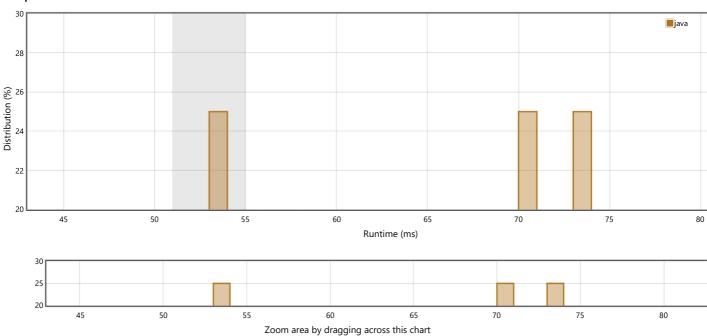
Runtime: 53 ms

Memory Usage: 108.4 MB

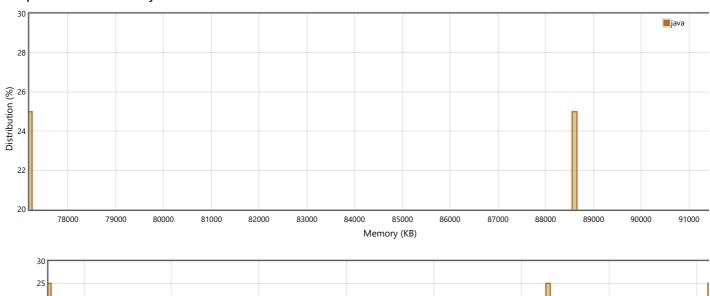
Status: Accepted

Submitted: 1 day ago

## **Accepted Solutions Runtime Distribution**



## **Accepted Solutions Memory Distribution**



84000

Zoom area by dragging across this chart

86000

88000

Invite friends to challenge Frequency Tracker

80000

82000

**√**5

78000

Submitted Code: 1 day ago

Language: java

Edit Code

92000

90000

```
2
          private Map<Integer, Integer> numToFreqMap; // map of number to frequency
          private Map<Integer, Integer> freqToCountMap; // map of frequency to count
 4
          public FrequencyTracker() {
 5
               numToFreqMap = new HashMap<>();
freqToCountMap = new HashMap<>();
 6
7
 8
9
10
11
          public void add(int number) {
               int freq = numToFreqMap.getOrDefault(number, 0);
               int newFreq = freq + 1;
numToFreqMap.put(number, newFreq);
12
13
                freqToCountMap.put(newFreq, freqToCountMap.getOrDefault(newFreq, 0) + 1);
15
               if (freq > 0) {
                    freqToCountMap.put(freq, freqToCountMap.get(freq) - 1);
if (freqToCountMap.get(freq) == 0) {
    freqToCountMap.remove(freq);
16
17
18
19
20
               }
21
          }
22
          public void deleteOne(int number) {
23
               int freq = numToFreqMap.getOrDefault(number, θ);
if (freq > θ) {
24
25
                    numToFreqMap.put(number, freq - 1);
freqToCountMap.put(freq, freqToCountMap.get(freq) - 1);
if (freqToCountMap.get(freq) == 0) {
26
27
28
                          freqToCountMap.remove(freq);
29
30
                    if (freq > 1) {
    freqToCountMap.put(freq - 1, freqToCountMap.getOrDefault(freq - 1, 0) + 1);
31
32
33
               }
35
          }
36
37
          public boolean hasFrequency(int frequency) {
38
               return freqToCountMap.containsKey(frequency);
39
40
```

Back to problem (/contest/weekly-contest-344/problems/frequency-tracker/)

Copyright © 2023 LeetCode

 $\begin{tabular}{lll} Help Center (/support) & | & Jobs (/jobs) & | & Bug Bounty (/bugbounty) & | & Online Interview (/interview/) & | & Students (/student) & | & Terms (/terms) & | & Privacy Policy (/privacy) & | & Priv$ 

United States (/region)