Interview Cake

You decide to test if your oddly-mathematical heating company is fulfilling its *All-Time Max, Min, Mean and Mode Temperature Guarantee*™.

Write a class `TempTracker` with these methods:

1. `insert()`—records a new temperature
2. `get_max()`—returns the highest temp we've seen so far
3. `get_min()`—returns the lowest temp we've seen so far
4. `get_mean()`—returns the mean

   > The **mean** of a set of values is the **average** value of all the items in the set.
   >
   > $$\text{mean} = \frac{\text{sum of all values}}{\text{number of values}}$$

   of all temps we've seen so far
5. `get_mode()`—returns a mode

   > The **mode** of a set of values is the **number which appears the most times**.
   >
   > For example, in this set:
   >
   > ```
   > 1, 3, 6, 3, 1, 3
   > ```
   >
   > The number 3 appears the most times, so it's the mode.
   >
   > **Careful:** a set may have multiple modes.

   of all temps we've seen so far

Optimize for space and time. **Favor speeding up the getter functions `get_max()`, `get_min()`, `get_mean()`, and `get_mode()` over speeding up the `insert()` function.**

`get_mean()` should return a **float**, but the rest of the getter functions can return **integers**. Temperatures will all be inserted as integers. We'll record our temperatures in Fahrenheit, so we can assume they'll all be in the range 0..110.

If there is more than one mode, return any of the modes.

Gotchas

We can get $O(1)$ time for all functions.

We can get away with only using $O(1)$ additional space. If you're storing each temperature as it comes in, be careful! You might be taking up $O(n)$ space, where n is the number of temperatures we insert!

Are you trying to be fancy about returning multiple modes if there's a tie? Good idea, but *read the problem statement carefully*! Check out that last sentence!

> Failing to carefully read or listen to the problem statement is a *very* common mistake, and it *always* looks bad. Don't let it happen to you.

Breakdown

Solution

This part requires full access (/upgrade)

You've already used up your free full access questions!

If you subscribe to our weekly newsletter (/free-weekly-coding-interview-problem-newsletter), you'll get another free full access question every week.

To see the full solution and breakdown for *all* of our questions, you'll have to upgrade to full access (/upgrade).

Upgrade now ➜ (/upgrade)

Complexity

$O(1)$ time for each function, and $O(1)$ space related to input! (Our `occurrences` list's size is bounded by our range of possible temps, in this case 0-110)

What We Learned

This part requires full access (/upgrade)

You've already used up your free full access questions!

If you subscribe to our weekly newsletter (/free-weekly-coding-interview-problem-newsletter), you'll get another free full access question every week.

To see the full solution and breakdown for *all* of our questions, you'll have to upgrade to full access (/upgrade).

Upgrade now ➜ (/upgrade)

Want more coding interview help?

Check out **interviewcake.com** for more advice, guides, and practice questions.