Module 8 Homework - Global Temperatures

The World Meteorological Organization (WMO) has hired you to write a data structure that stores record temperatures across the globe.

Incoming temperature reports have very precise geo-location data (latitude and longitude coordinates). The WMO would like for you to store data in a "grid", treating any points that round to the same latitude and longitude within one decimal as the same.

```
>>> p1 = (36.4482, -105.0072) # 4 decimal point precision (lat, long)
>>> p2 = (36.4355, -105.0388) # the same "grid point" as above
```

Note that we are using negative numbers for southern latitudes and western longitudes, so the above coordinates correspond to (36.4° N, 105.0° W).

Write a data structure `TempMap` that keeps track of the hottest and coldest temperatures for each reported coordinate. There is a lot of data coming in, so you need to read and write this information quickly - you should be able to update the record at a (lat, long) pair in *O*(1).

Do not use the built-in set or dictionary types in this assignment.

class `TempMap`

Write a class `TempMap` that keeps track of the maximum and minimum temperature for each reported set of coordinates.

Magic Methods

- `init`
    - *O*(1)
- `in`
    - *O*(1)
    - Returns `True` (`False`) if the rounded version of a specified coordinate is (is not) in this collection
- `get`
    - *O*(1)
    - Returns a tuple containing the (min, max) temperatures at these coordinates
    - Raise a `KeyError` if the specified coordinate are not in this collection
    - the input coordinates may or may not be rounded

Non-Magic Methods

- `add_report(pos, temp)`

    - *O*(1)
    - pos - an (unrounded) tuple of coordinates
    - temp - the current temperature at those coordinates
    - Updates the maximum or minimum recorded temperature for pos if appropriate

- remove_pos(pos)

    - *O*(1)
    - removes the record for the given position from this collection
    - raises a KeyError if pos is not in this collection

Special Behavior

- Memory constraint - you should limit your memory usage to be between 1/2 and 2x the amount necessary to store the number of unique coordinates - that is, you should periodically increase and decrease the amount of memory used as items are added and removed.

- The starter code includes a function generate_report that returns a randomly generated tuple of (lat, long, temperature). Feel free to use it to help with testing.

Examples

Any examples below are intended to be illustrative, not exhaustive. Your code may have bugs even if it behaves as below. Write your own tests, and think carefully about edge cases.

```
>>> from TempMap import *
>>> tm = TempMap()
>>> p1 = (36.4482, -105.0072)
>>> p2 = (36.44, -105.00)     # rounds to the same as p1
>>> p3 = (36.45, -105.00)     # rounds to different than p1
>>> # Examples: `in` and `add_report`
>>> p1 in tm
False
>>> tm.add_report(p1, 25)
>>> p1 in tm
True
>>> p2 in tm
True
>>> p3 in tm
False
>>> # Examples: `get` and `add_report`
>>> tm[p1]
(25, 25)
>>> tm.add_report(p1, 25.7) # new high
>>> tm[p1]
(25, 25.7)
>>> tm.add_report(p1, 20)   # new low
>>> tm[p1]
(20, 25.7)
>>> tm.add_report(p1, 24)   # no update
>>> tm[p1]
```

```
(20, 25.7)
>>> tm[p3]                  # no entry
Traceback (most recent call last):
...
KeyError: 'No records for pos (36.5, -105.0).'
>>> # Examples: `remove_pos`
>>> tm.remove_pos(p1)
>>> tm[p1]                 # key error - get
Traceback (most recent call last):
...
KeyError: 'No records for pos (36.4, -105.0).'
>>> tm.remove_pos(p1)      # key error - remove
Traceback (most recent call last):
...
KeyError: 'No records for pos (36.4, -105.0).'
```

Submission

At a minimum, submit the following file with the class described above.

- `TempMap.py`

Students must submit to Mimir **individually** by the due date (typically, the second Wednesday after this module opens at 11:59 pm EST) to receive credit.

Grading

If you use the built-in set or dictionary to implement functionality, you will recieve a 0 on this assignment.

It is okay to use the set or dictionary for testing purposes only.

- 30 - functionality
    - 10 - `add_report`/`in`
    - 10 - `get`
    - 10 - `remove`
- 50 - $O(1)$ Running Times
    - 25 - `add_report` and `get`
    - 25 - `in` and `remove`
- 20 - Memory Management (manually graded)

Feedback

If you have any feedback on this assignment, please leave it here.

We check this feedback regularly. It has resulted in:

- A simplified, clear **Submitting** section on all assignments
- A simplified, clear **Grading** section on all assignments

- Clearer instructions on assignments