Mod 8 Lab: Custom Set

Description

Impliment a class `CustomSet` that mirrors the python set. **Do not use the built-in `set` or `dict` classes in this assignment.**

Magic Metohds

- *O*(1): `len` - Returns the number of items in `CustomSet`
- *O*(1): `contains` - Returns `True` (`False`) if an object is (is not) in `CustomSet`

Normal Methods

- *O*(1): `add(item)` - Adds an item to `CustomSet`.
- *O*(1): `remove(item)` - Removes item from `CustomSet`

Special Cases

- `add` - If the specified item is already in the set, do nothing: you should not add duplicate items, and you should not raise any errors
- `remove` - Raise a `ValueError` if a user tries to delete an object not in `CustomSet`
- Your class should support any hashable object, including strings

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 CustomSet import CustomSet
>>> s = CustomSet()
>>> "hello" in s
False
>>> s.add("hello")
>>> len(s)
1
>>> "hello" in s
True
>>> s.remove("hello")
>>> s.remove(5)
Traceback (most recent call last):
...
ValueError: Attempt to remove non-extant item 5
>>>
```

Submitting

At a minimum, submit a file named `CustomSet.py` containing a class `CustomSet`.

Students must submit to Mimir **individually** by the due date (typically, two days after lab at 11:59 pm EST) to receive credit.

Grading

You will recieve a 0 on this assignment if you use the built-in `set` or `dict` types to impliment functionality for CustomSet.

- 30 - `add`
  - 5 - Functionality
  - 25 - Speed
- 35 - `remove`
  - 5 - Functionality
  - 5 - Special Case: `ValueError`
  - 25 - Speed
- 30 - `contains`
  - 5 - Functionality
  - 25 - Speed
- 5 - `len`
  - 5 - Functionality

In Mimir, some tests for `add` and `contains` are combined (e.g. 10 points for "add/contains" functionality.

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 several assignments (particularly in the recursion module)