Coding Project II

Find x in Infinite Array - 10 Points

In this assignment you will implement a working solution to [DPV] 2.16:

> You are given an infinite array $A[\cdot]$ in which the first n cells contain integers in sorted order and the rest of the cells are filled with ∞. You are not given the value of n. Describe an algorithm that takes an integer x as input and finds a position in the array containing x, if such a position exists, in $O(\log n)$ time.

Your code is prohibited from directly accessing $A[\cdot]$ - the template provides the following:

- **findX.start(seed, nLower, nUpper)** - initialize the problem space and return the value x
- **findX.lookup(index)** - return the value of $A[index]$, or ***None*** if $index > n$
- **findX.lookups()** - return the number of calls to **findX.lookup()**

Please note that the range of values for $A[\cdot]$ is $[1..\infty]$, and that x is guaranteed to exist in $A[\cdot]$ The number of calls to findX.lookup() will be strictly limited. Exceeding that limit will throw an exception. The successful execution of your solver should print the index where x is found within $A[\cdot]$ and the number of lookups required to find x. Here is the base case to validate your work:

```
findX result: x found at index 10759 in 32 calls
```

You may also change the random seed to evaluate your algorithm:

```
$python3 findX.py -s 123456
```

Restrictions

- You must complete this assignment on your own; do not share your code with anyone and do not copy code from the Internet
- You must be fully compatible with **python (3.6.x)**
- No additional libraries may be imported beyond what is provided in the assignment
- Do not modify the structure or program-flow of this assignment in any way – only add code where directed to do so by the code comments. Do not add functions, variables, or other code constructions except where told to do so.

Rubric

For each test case, you will receive 1 point for finding the correct index and a second point if done within the expected number of calls ($\pm$ 2) - the expected number is based upon the position of x in $A[]$ and not necessarily the strict upper limit based on n. The Autograder will confirm your submission against the base case. Your solution will be tested against four additional cases for a total of 10 possible points.

Submission

Submit your code file (**findX.py**) ONLY to the Gradescope assignment on or before the posted due date. Do not submit a zip file, or any other files but **findX.py**. Late submissions will not be accepted.