Coding Project II

Find x in Infinite Array - 10 Points

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

> You are given an infinite array $A[\cdot]$ in which the first n cells contain integers in sorted non-descending order and the rest of the cells are filled with `None`. 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 $A[\cdot]$ - you may use the following methods:

- **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 $n > 0$, the index range for A is $A[1, \ldots, \infty]$, the numbers in A may repeat, and that x may ***not*** exist in $A[\cdot]$. If x does exist it is guaranteed to be unique. The successful execution of your solver should return the index where x is found within $A[\cdot]$, or `None` if x does not exist, as well as the number of lookups. Running your solution against the base case (seed=1234, nLower=10, nUpper=100000) should produce:

```
findX result:  x=134783 found at index 48335 in 32 calls
```

Your implementation may make fewer calls - this is OK. You may also change the settings to evaluate your algorithm:

```
$python3 findX.py -s 123456 -l 100000 -u 200000
```

Your program is allowed to make at most $((2 \times \log n) + 2)$ calls to `findX.lookup`, at which point an exception will be raised. This upper limit is used to protect against infinite loops.

Restrictions

- You must complete this assignment on your own; do not share your code with anyone and do not copy code from the Internet.
- Template code is provided and must be used. ***Note: the templates are refreshed each semester. Please be sure to use the current version.***
- You code must be compatible with **python 3.10**.
- No additional libraries may be imported beyond what is provided in the assignment template.
- 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 – each individual component of your submission will be tested by the auto-grader when it is submitted.

Rubric

For each test case, you will receive 1 point for finding the correct index and a second point if done within an upper limit of the ***expected*** number of calls - the ***expected*** number is based upon the position of x in $A[\cdot]$ and not necessarily the strict upper limit described above. 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.