2336. Smallest Number in Infinite Set

Description

You have a set which contains all positive integers `[1, 2, 3, 4, 5, ...]`.

Implement the `SmallestInfiniteSet` class:

- `SmallestInfiniteSet()` Initializes the **SmallestInfiniteSet** object to contain **all** positive integers.
- `int popSmallest()` **Removes** and returns the smallest integer contained in the infinite set.
- `void addBack(int num)` **Adds** a positive integer `num` back into the infinite set, if it is **not** already in the infinite set.

Example 1:

```
 Input
["SmallestInfiniteSet", "addBack", "popSmallest", "popSmallest", "popSmallest", "addBack", "popSmallest", "popSmallest", "popSmallest"]
[[], [2], [], [], [], [1], [], [], []]
 Output
[null, null, 1, 2, 3, null, 1, 4, 5]

 Explanation
SmallestInfiniteSet smallestInfiniteSet = new SmallestInfiniteSet();
smallestInfiniteSet.addBack(2);    // 2 is already in the set, so no change is made.
smallestInfiniteSet.popSmallest(); // return 1, since 1 is the smallest number, and remove it from the set.
smallestInfiniteSet.popSmallest(); // return 2, and remove it from the set.
smallestInfiniteSet.popSmallest(); // return 3, and remove it from the set.
smallestInfiniteSet.addBack(1);    // 1 is added back to the set.
smallestInfiniteSet.popSmallest(); // return 1, since 1 was added back to the set and
                                   // is the smallest number, and remove it from the set.
smallestInfiniteSet.popSmallest(); // return 4, and remove it from the set.
smallestInfiniteSet.popSmallest(); // return 5, and remove it from the set.
```

Constraints:

- `1 <= num <= 1000`
- At most `1000` calls will be made **in total** to `popSmallest` and `addBack`.