lee215 ★ 47721 Last Edit: June 28, 2019 9:11 AM 2.8K VIEWS

34

Intuition

`A[i]` is distinct and ascending.
`A[i] - i` is non-descending array.
Binary search the first `0` in the array of `A[i] - i`.

Easy prove

`A[i] < A[i + 1]`
`A[i] <= A[i + 1] - 1`
`A[i] - i <= A[i + 1] - i - 1`

Test Cases

It said that "return the smallest index i that satisfies A[i] == i".
Many solution just "return an whatever index that satisfies A[i] == i".
Tough it can get accepted.

Update

I used to return any index that `A[i] == i`.
It got a false accepted due to weak test cases.
Thanks to @dibdidib for reminding me and now I fixed it.

Complexity

Time `O(logN)`, Space `O(1)`

Java:

```
public int fixedPoint(int[] A) {
    int l = 0, r = A.length - 1;
    while (l < r) {
        int m = (l + r) / 2;
        if (A[m] - m < 0)
            l = m + 1;
        else
            r = m;
    }
    return A[l] == l ? l : -1;
}
```

C++:

```
int fixedPoint(vector<int>& A) {
    int l = 0, r = A.size() - 1, m;
    while (l < r) {
        m = (l + r) / 2;
        if (A[m] - m < 0)
            l = m + 1;
        else
            r = m;
    }
    return A[l] == l ? l : -1;
}
```

Python:

```
def fixedPoint(self, A):
    l, r = 0, len(A) - 1
    while l < r:
        m = (l + r) / 2
        if A[m] - m < 0:
            l = m + 1
        else:
            r = m
    return l if A[l] == l else -1
```