

## Progress

[Course](#) > [Readings/Videos](#) > [Reading 4: Specifications](#) > [Questions](#)

[illegible]

## Questions

 [Bookmark this page](#)

behave nicely

1/1 point (graded)

```
static int findFirst(int[] a, int val) {
    for (int i = 0; i < a.length; i++) {
        if (a[i] == val) return i;
    }
    return a.length;
}
```

```
static int findLast(int[] a, int val) {
    for (int i = a.length - 1; i >= 0; i--) {
        if (a[i] == val) return i;
    }
    return -1;
}
```

If clients only care about calling the find method when they know `val` occurs exactly once in `a`, are `findFirst` and `findLast` behaviorally equivalent?

☒ Yes

☐ No



**Explanation**

If `val` occurs exactly once in `a`, then it doesn't matter whether we search from start to end or end to start; and it doesn't matter what we do when we don't find `val`.

Once we define how specifications are structured, we'll see that they are equivalent in this case because a strong precondition hides their potential differences in behavior.

Submit

Show Answer

 Answers are displayed within the problem

best behavior

1/1 point (graded)

Suppose clients only care that the find method should return:

- any index `i` such that `a[i] == val`, if `val` is in `a`

- any integer `j` such that `j` is not a valid array index, otherwise

In this case, are `findFirst` and `findLast` behaviorally equivalent?

☒ Yes

☐ No



### Explanation

Both specifications satisfy these very minimal requirements on the return value.

Once we define how specifications are structured, we'll see that they are equivalent in this case because a weak postcondition permits their differences in behavior.

Submit Show Answer

**i** Answers are displayed within the problem

[← Previous](#)
[Next →](#)

Open Learning Library

About

Accessibility

All Courses

Why Support MIT Open Learning?

Help

Connect

Contact

Twitter

Facebook

Privacy Policy

Terms of Service

© Massachusetts Institute of Technology, 2024