Questions

Bookmark this page

distinguished

2/2 points (graded)

We just saw the following implementations:

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

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

Now consider this possible specification of `find`:

```
static int find(int[] arr, int val)
  requires: nothing
  effects: returns largest index i such that
             arr[i] = val, or -1 if no such i
```

Which input(s) demonstrate that `findFirst` does not satisfy this spec? Check all that apply.

- [x] `[ 1, 2, 2 ], 2`
- [ ] `[ 1, 2, 3 ], 2`
- [x] `[ 1, 2, 3 ], 4`
- [ ] none of the above, `findFirst` does satisfy this spec!

✔

Which input(s) demonstrate that `findLast` does not satisfy this spec? Check all that apply.

- [ ] `[ 1, 2, 2 ], 2`
- [ ] `[ 1, 2, 3 ], 2`

- [ ] `[ 1, 2, 3 ], 4`
- [x] none of the above, `findLast` does satisfy the spec!

✔

Submit

Show Answer

✔ Correct (2/2 points)

over/under (a)

1/1 point (graded)

Previously we saw this *under-determined* version of the *find* specification: (let's call it *findUnderdet*)

```
static int findUnderdet(int[] arr, int val)
  requires: val occurs in arr
  effects:  returns index i such that arr[i] = val
```

This spec allows multiple possible return values for arrays with duplicate values.

For each spec below, say whether it is *less* fully-determined than *findUnderdet*, *more* fully-determined, or deterministic.

```
static int find(int[] arr, int val)
  requires: val occurs exactly once in arr
  effects:  returns index i such that arr[i] = val
```

- (●) deterministic
- ( ) more fully-determined than *findUnderdet*, but not deterministic
- ( ) less fully-determined than *findUnderdet*

✔

Explanation
This is our original, deterministic spec.

Submit

Show Answer

ℹ Answers are displayed within the problem

over/under (b)

1/1 point (graded)

```
static int find(int[] arr, int val)
  requires: nothing
  effects: returns largest index i such that arr[i] = val, or -1 if no such i
```

- (●) deterministic
- ( ) more fully-determined than *findUnderdet*, but not deterministic
- ( ) less fully-determined than *findUnderdet*

✔

Explanation
This is our version of the spec that distinguishes the two implementations in the reading. It is also determinstic.

Submit

Show Answer

Answers are displayed within the problem

over/under (c)

1/1 point (graded)

```
static int find(int[] arr, int val)
  requires: val occurs in arr
  effects: returns largest index i such that arr[i] = val
```

- (•) deterministic
- ( ) more fully-determined than *findUnderdet*, but not deterministic
- ( ) less fully-determined than *findUnderdet*

✔

Explanation
Once again, the spec is fully-determined.

Submit

Show Answer

Answers are displayed within the problem

Previous Next