Array Slicing

Array slicing involves taking a subset from an array and **allocating a new array with those elements**.

In JavaScript you can create a new array of the elements in `myArray`, from `startIndex` to `endIndex` (exclusive), like this:

```JavaScript
myArray.slice(startIndex, endIndex);
```

You can also get everything *after* `startIndex` by just omitting `endIndex`:

```JavaScript
myArray.slice(startIndex);
```

Careful: there's a hidden time and space cost here! It's tempting to think of slicing as just "getting elements," but in reality you are:

1. allocating a new array
2. *copying* the elements from the original array to the new array

This takes $O(n)$ time and $O(n)$ space, where n is the number of elements in the *resulting* array.

That's a bit easier to see when you save the result of the slice to a variable:

```JavaScript
var tailOfArray = myArray.slice(1);
```

But a bit harder to see when you don't save the result of the slice to a variable:

```JavaScript
return myArray.slice(1);
// whoops, I just spent O(n) time and space!
```

```JavaScript
myArray.slice(1).forEach(function(item) {
    // whoops, I just spent O(n) time and space!
});
```

So keep an eye out. Slice wisely.

See also:

- Arrays (/concept/array)
- In-Place Algorithm (/concept/in-place)

What's next?

If you're ready to start applying these concepts to some problems, check out our mock coding interview questions (/next).

They mimic a real interview by offering hints when you're stuck or you're missing an optimization.

Try some questions now ➜
