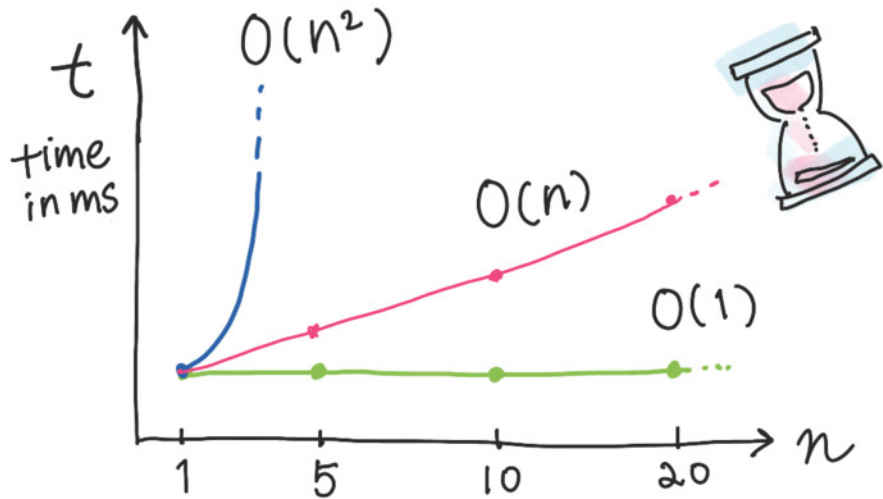# Big O Notation

JS Let's see if we have some function that doesn't actually loop the array:

```js
const alwaysTrueNoMatterWhat = (haystack) => {
    return true;
```

n=5
n=10
n=20
⋮

↜ Array size has no effect on the runtime

⭐ Constant time

||

$O(1)$

$O(n^2)$

t

time in ms

$O(n)$

$O(1)$

1   5   10   20 n

⭐ Quadratic time = $O(n^2)$

n=5, however the runtime is proportional to n^2

```js
const
array2 = [🐱, 🌸, 💩, 💩, 🖊 ];
```

JS
```js
const hasDuplicates = (arr) => {
    for (let i=0; i < arr.length; i++)
        let item = arr[i];
        if(arr.slice(i+1).indexOf(item) !== -1) {
            return true;
        }
    return false;
}
```

① Loop thru the array

② Another array lookup w/ indexOf method