

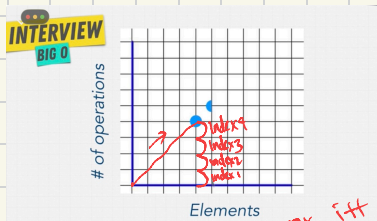
# DSA:

scalable



misal isi array  
["A", "B", "C"]

$O(n)$   
Linear time



for  $i=0$   $i < \text{length array}$   $i++ = O(n)$

```
const memo = ['memo'];
const everyone = ['dory', 'bruce', 'marlin', 'nemo', 'gill', 'bloat', 'nigel', 'squirt', 'daria', 'hank'];
const large = new Array(100000).fill('memo');

function findMemo(array) {
  for (let i = 0; i < array.length; i++) {
    if (array[i] === 'memo') {
      console.log('Found MEMO!');
    }
  }
}

findMemo(large); // 0.100000 -> Linear Time
```

**INTERVIEW BIG O**

```
//ES5
function compressAllBoxes(boxes) {
  boxes.forEach(function(item) {
    console.log(item);
  });
}

//ES6
const compressAllBoxes = boxes => {
  boxes.forEach(box => console.log(box));
}
```

**INTERVIEW BIG O**

```
//ES5
function compressAllBoxes(boxes) {
  boxes.forEach(function(item) {
    console.log(item);
  });
}

//ES6
const compressAllBoxes = boxes => {
  boxes.forEach(box => console.log(box));
}
```

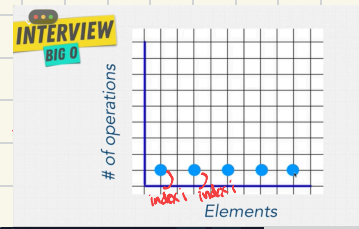
$O(n)$  - Linear time

$O(1)$  - Constant Time

**INTERVIEW BIG O**

```
function compressFirstBox(boxes) {
  console.log(boxes[0]);
}
```

$O(1)$  - Constant Time



```
const boxes = ['A', 'B', 'C', 'D', 'E'];

function logFirstTwoBoxes(boxes) {
  console.log(boxes[0]); // 0
  console.log(boxes[1]); // 1
}

logFirstTwoBoxes(boxes); // 0.000000 -> O(1)
```

