Multiplying Runtimes

The big idea:

If your algorithm is in the form “do this for each time you do that” then you multiply the runtimes.

— *Cracking the Coding Interview*

```
1   logTwoArrays(["A", "B", "C", "D"], [1, 2, 3, 4])
2
3   function logTwoArrays(arrA, arrB){
4     for(let i=0; i<arrA.length; i++){
5       console.log(arrA[i])
6       for(let j=0; j<arrB.length; j++){
7         console.log(arrB[j])
8       }
9     }
10  }
11
```

In this example, we iterate through arrB for every item in arrA, so the runtimes are multiplied together making the Big O Notation O(arrA * arrB), which may be abstracted as O(N * M)

Heads up!

We do not drop terms from different data sets when we are not aware of the data they contain, because we are unable to ascertain if they will be dominant.

In this example, we iterate through arrA for every item in arrA. The runtimes are multiplied together making the Big O Notation O(arrA * arrA) or more simply O($arrA^2$) (abstracted as O(N^2), *then* we add the runtime of arrB making the final Big O notation of this function O($arrA^2$ + arrB) which may be abstracted as O (N^2 + M)

```
3   function logTwoArrays(arrA, arrB){
4     for(let i=0; i<arrA.length; i++){
5       console.log(arrA[i])
6       for(let j=0; j<arrA.length; j++){
7         console.log(arrA[j])
8       }
9     }
10    for(let j=0; j<arrB.length; j++){
11      console.log(arrB[j])
12    }
13  }
```