Big-O Notation Practice

In this exercise, you'll analyze expressions and code to figure out the time complexity.

Step One: Simplifying Expressions

Simplify the following big O expressions as much as possible:

```
1. O(n + 10) O(n)

2. O(100 * n) O(n)

3. O(25) O(n)

4. O(n^2 + n^3) O(n^3)

5. O(n + n + n + n) O(n)

6. O(1000 * \log(n) + n)

7. O(1000 * n * \log(n) + n)

8. O(2^n + n^2) O(2^n)

9. O(5 + 3 + 1) O(1)

10. O(n + n^n(1/2) + n^2 + n * \log(n)^10) O(n^3)
```

Step Two: Calculating Time Complexity

Determine the time complexities for each of the following functions. If you're not sure what these functions do, copy and paste them into the console and experiment with different inputs!

```
function logUpTo(n) {
  for (let i = 1; i <= n; i++) {
    console.log(i);
  }
}</pre>
```

Time Complexity:

Time Complexity:

```
function logAtMost10(n) {
  for (let i = 1; i <= Math.min(n, 10); i++) {
    console.log(i);
  }
}</pre>
```

Time Complexity:

```
function onlyElementsAtEvenIndex(array) {
  let newArray = [];
  for (let i = 0; i < array.length; i++) {
    if (i % 2 === 0) {
      newArray.push(array[i]);
    }
  }
  return newArray;
}</pre>
```

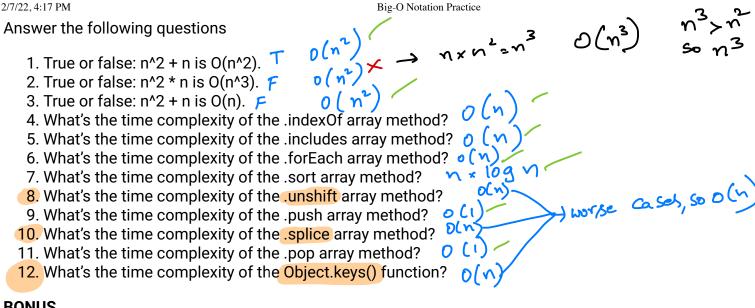
Time Complexity:

```
function subtotals(array) {
  let subtotalArray = [];
  for (let i = 0; i < array.length; i++) {
    let subtotal = 0;
    for (let j = 0; j <= i; j++) {
        subtotal += array[j];
    }
    subtotalArray.push(subtotal);
}
return subtotalArray;
}</pre>
```

Time Complexity:

Time Complexity:

Part 3 - short answer



BONUS

13. What's the space complexity of the Object.keys() function?

Solution

View our solutions <solution/index.html>