Simplifying Expressions

1 ) O(n)

2) O(n)

3) O(1)

4) O(n^3)

5) O(n)

6) O(n)

7) O(n log n)

8) O(2^ n)

9)O(1)

10)O(n^2)

Calculating Time Complexity

function logUpTo(n) {

for (let i = 1; i <= n; i++) {

console.log(i);

}

}

O(n)

1. function logAtLeast10(n) { for (let i = 1; i <= Math.max(n, 10); i++) { console.log(i); } }

O(n)

1. function logAtMost10(n) {

for (let i = 1; i <= Math.min(n, 10); i++) {

console.log(i);

}

}

O(1)

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

O(n)

1. 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; }

O(n^2)

1. function vowelCount(str) {

let vowelCount = {};

const vowels = "aeiouAEIOU";

for (let char of str) {

if(vowels.includes(char)) {

if(char in vowelCount) {

vowelCount[char] += 1;

} else {

vowelCount[char] = 1;

}

}

}

return vowelCount;

}

O(n)

Short Answer

1. True
2. True
3. False
4. O(n)
5. O(n)
6. O(n)
7. O(n log n)
8. O(n)
9. O(1)
10. O(n)
11. O(1
12. O(n)
13. O(n)