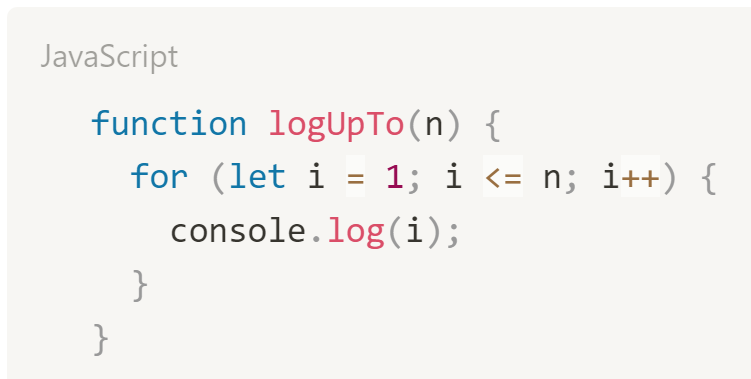
**Step One: Simplifying Expressions**

Simplify the following big O expressions as much as possible:

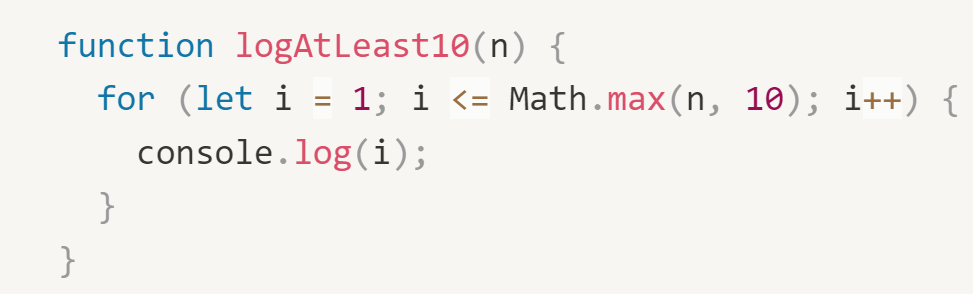
1. O(n + 10) 🡪 O(n)
2. O(100 \* n) 🡪 O(n)
3. O(25) 🡪 O(1)
4. O(n^2 + n^3) 🡪 O(n^3)
5. O(n + n + n + n) 🡪 O(n)
6. O(1000 \* log(n) + n) 🡪 O(n)
7. O(1000 \* n \* log(n) + n) 🡪 O(n\*log(n))
8. O(2^n + n^2) 🡪 O(2^n)
9. O(5 + 3 + 1) 🡪 O(1)
10. O(n + n^(1/2) + n^2 + n \* log(n)^10) 🡪 O(n^2)

## ****Step Two: Calculating Time Complexity****

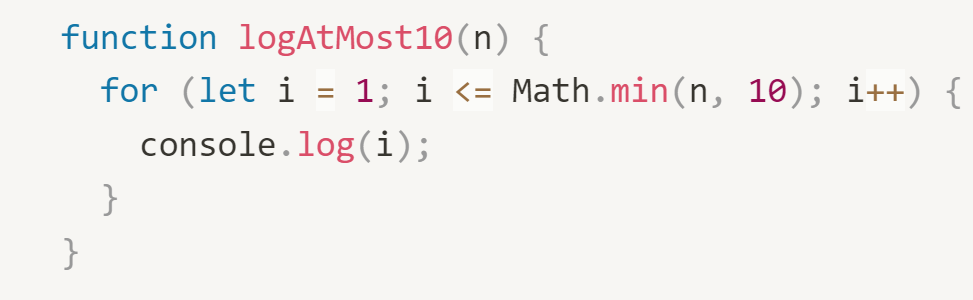
Determine the time complexities for each of the following functions. If you’re not sure what these functions do, copy and paste them into the console and experiment with different inputs!



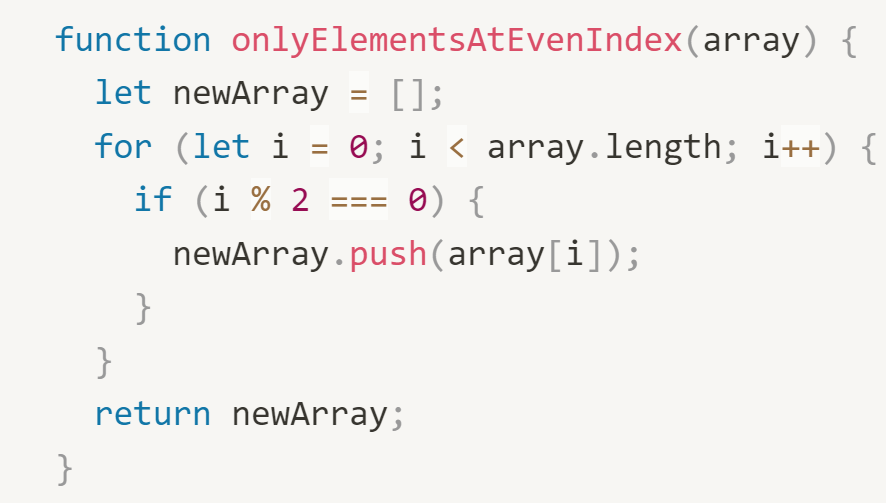
Time Complexity: O(n)



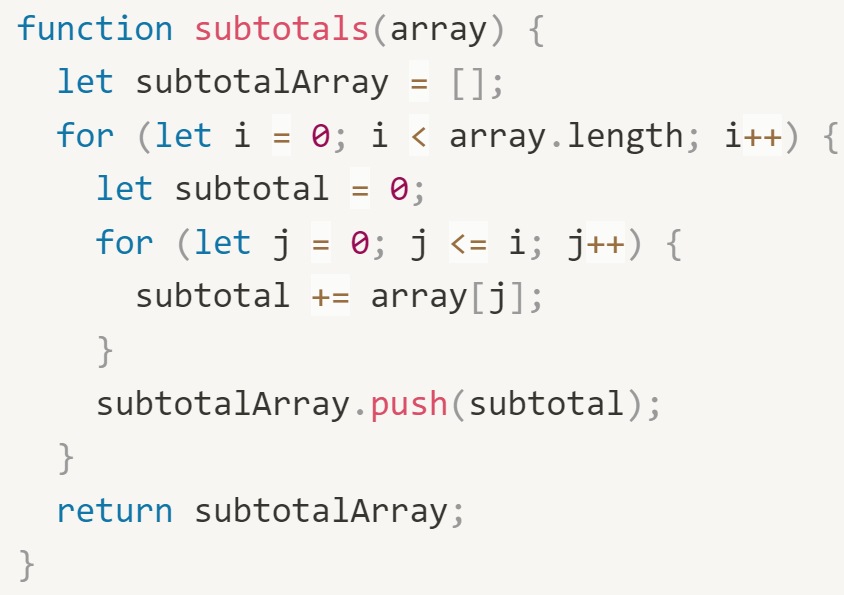
Time Complexity: O(n)



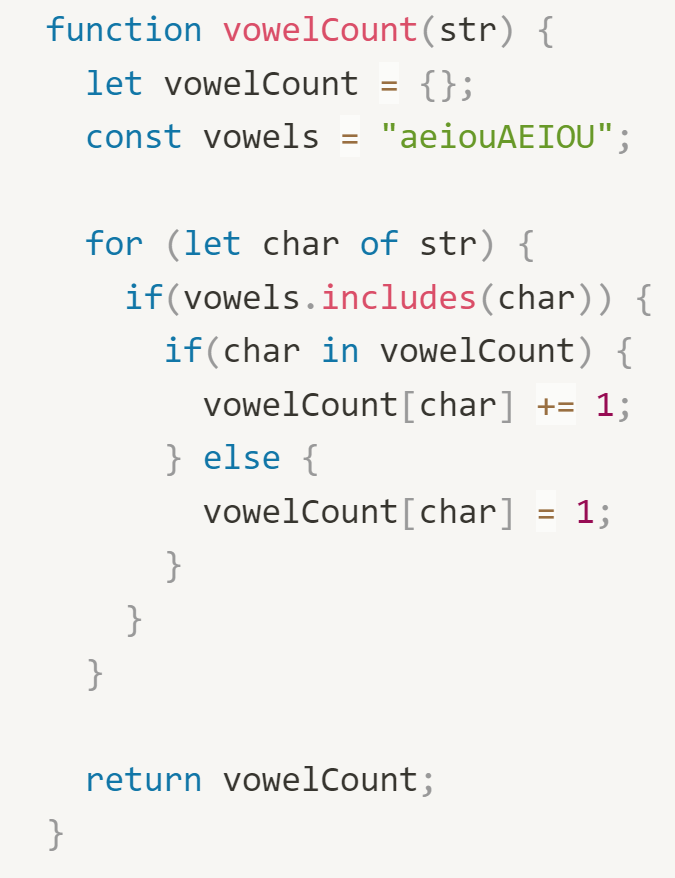
Time Complexity: O(1)



Time Complexity: O(n)



Time Complexity: O(n^2)



Time Complexity: O(n)

## ****Part 3 - short answer****

Answer the following questions

1. True or false: n^2 + n is O(n^2).

True

1. True or false: n^2 \* n is O(n^3).

True

1. True or false: n^2 + n is O(n).

False

1. What’s the time complexity of the .indexOf array method?

O(n)

1. What’s the time complexity of the .includes array method?

O(n)

1. What’s the time complexity of the .forEach array method?

O(n)

1. What’s the time complexity of the .sort array method?

O(n\*log(n))

1. What’s the time complexity of the .unshift array method?

O(n) 🡪 Need to shift indices of all existing elements in the array.

1. What’s the time complexity of the .push array method?

O(1)

1. What’s the time complexity of the .splice array method?

O(n) 🡪 Worse case scenario, insert an element at the beginning, then all indices of existing elements need to be updated.

1. What’s the time complexity of the .pop array method?

O(1)

1. What’s the time complexity of the Object.keys() function?

O(n)

### **BONUS**

1. What’s the space complexity of the Object.keys() function?

O(n)