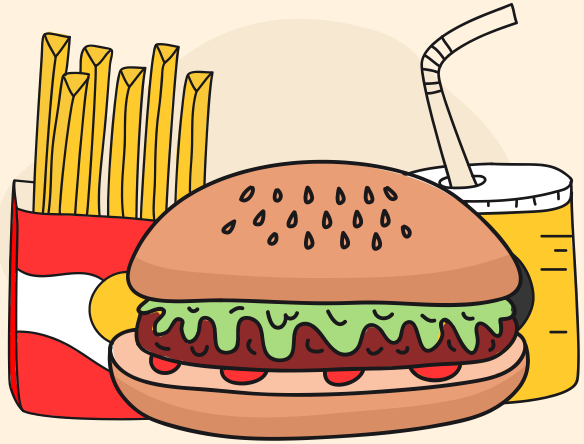


MORNING STANDEES

FRIDAY MARCH 4, 2022

TODAY'S SCHEDULE!!

9:00AM - 9:05AM - morning standees!!
9:05AM - 9:15AM - hack hour solution: get-all-products
9:15AM - 10:00AM - hack hour: max-subarray
10:00AM - 8:00PM - open source product time!!
12:00PM - 12:30PM - lecture: github cleaning
12:30PM - 1:30PM - LUNCH!!!!!!!!!!!!!!!!!
1:30PM - 1:35PM - midday standees!!
1:35PM - 2:00PM - mentor/mentee session
5:15PM - 5:30PM - tech talks round 1
5:30PM - 6:30PM - DINNNERRRRRR!!!!!!!
6:30PM - 6:45PM - tech talks round 2
10:00PM - FIRST DRAFT OF RESUMES ARE DUE!!!

HACK HOUR SOLUTION:
get-all-products

INPUT

OUTPUT

Given an array of at least two integers (which may be positive, negative, or zero), return an array of all the possible products made by multiplying all but one number. In other words, find all the products of multiplying any array.length - 1 numbers in the array.

ARRAY = [1,7,3,4]

```
const getAllProducts = array => {
    let zeroCount = 0;
    let products = 1;
    let indexAtZero;
    const zeroArray = new Array(array.length).fill(0);

    for (let i = 0; i < array.length; i++) {
        if (array[i] === 0) {
            zeroCount++;
            indexAtZero = i;
            if (zeroCount === 2) break;
        }
        else {
            products *= array[i];
        }
    }

    if (zeroCount !== 2) {
        if (zeroCount === 0) {
            return array.map(element => products / element);
        } else {
            zeroArray[indexAtZero] = products;
            return zeroArray
        }
    } else {
        return zeroArray;
    }
}
```

i = 0 array[0] = 1 zeroCount = 0

products = 1 × 1 = 1 $\frac{84}{1} = 84$

i = 1 array[1] = 7 zeroCount = 0

products = 1 × 7 = 7 $\frac{84}{7} = 12$

i = 2 array[2] = 3 zeroCount = 0

products = 7 × 3 = 21 $\frac{84}{3} = 28$

i = 3 array[3] = 4 zeroCount = 0

products = 21 × 4 = 84 $\frac{84}{4} = 21$

[84, 12, 28, 21]

ARRAY = [2,0,5]

```
const getAllProducts = array => {
    let zeroCount = 0;
    let products = 1;
    let indexAtZero;
    const zeroArray = new Array(array.length).fill(0);

    for (let i = 0; i < array.length; i++) {
        if (array[i] === 0) {
            zeroCount++;
            indexAtZero = i;
            if (zeroCount === 2) break;
        }
        else {
            products *= array[i];
        }
    }

    if (zeroCount !== 2) {
        if (zeroCount === 0) {
            return array.map(element => products / element);
        } else {
            zeroArray[indexAtZero] = products;
            return zeroArray
        }
    } else {
        return zeroArray;
    }
}
```

i = 0 array[0] = 2 zeroCount = 0

products = 1x2 = 2 indexAtZero =

i = 1 array[1] = 0 zeroCount = 1

products = 2 indexAtZero = 1

i = 2 array[2] = 5 zeroCount = 1

products = 2x5 = 10 indexAtZero = 1

[0, 10, 0]

ARRAY = [0, 48, 0]

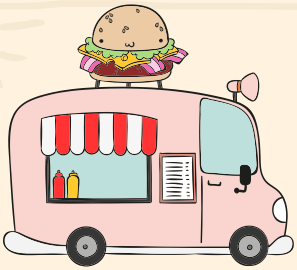
```
const getAllProducts = array => {
    let zeroCount = 0;
    let products = 1;
    let indexAtZero;
    const zeroArray = new Array(array.length).fill(0);

    for (let i = 0; i < array.length; i++) {
        if (array[i] === 0) {
            zeroCount++;
            indexAtZero = i;
            if (zeroCount === 2) break;
        }
        else {
            products *= array[i];
        }
    }

    if (zeroCount !== 2) {
        if (zeroCount === 0) {
            return array.map(element => products / element);
        } else {
            zeroArray[indexAtZero] = products;
            return zeroArray
        }
    } else {
        return zeroArray;
    }
}
```

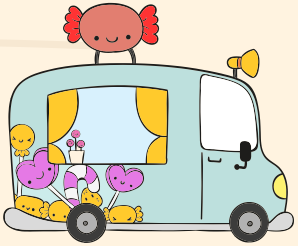
i = 0 array[0] = 0 zeroCount = 1
products = 1 indexAtZero = 0

i = 1 array[1] = 48 zeroCount = 1
products = 1x48 = 48 indexAtZero = 0

i = 2 array[2] = 0 zeroCount = 2
products = indexAtZero = 2

[0, 0, 0]





HACK HOUR CHALLENGE:
max-subarray