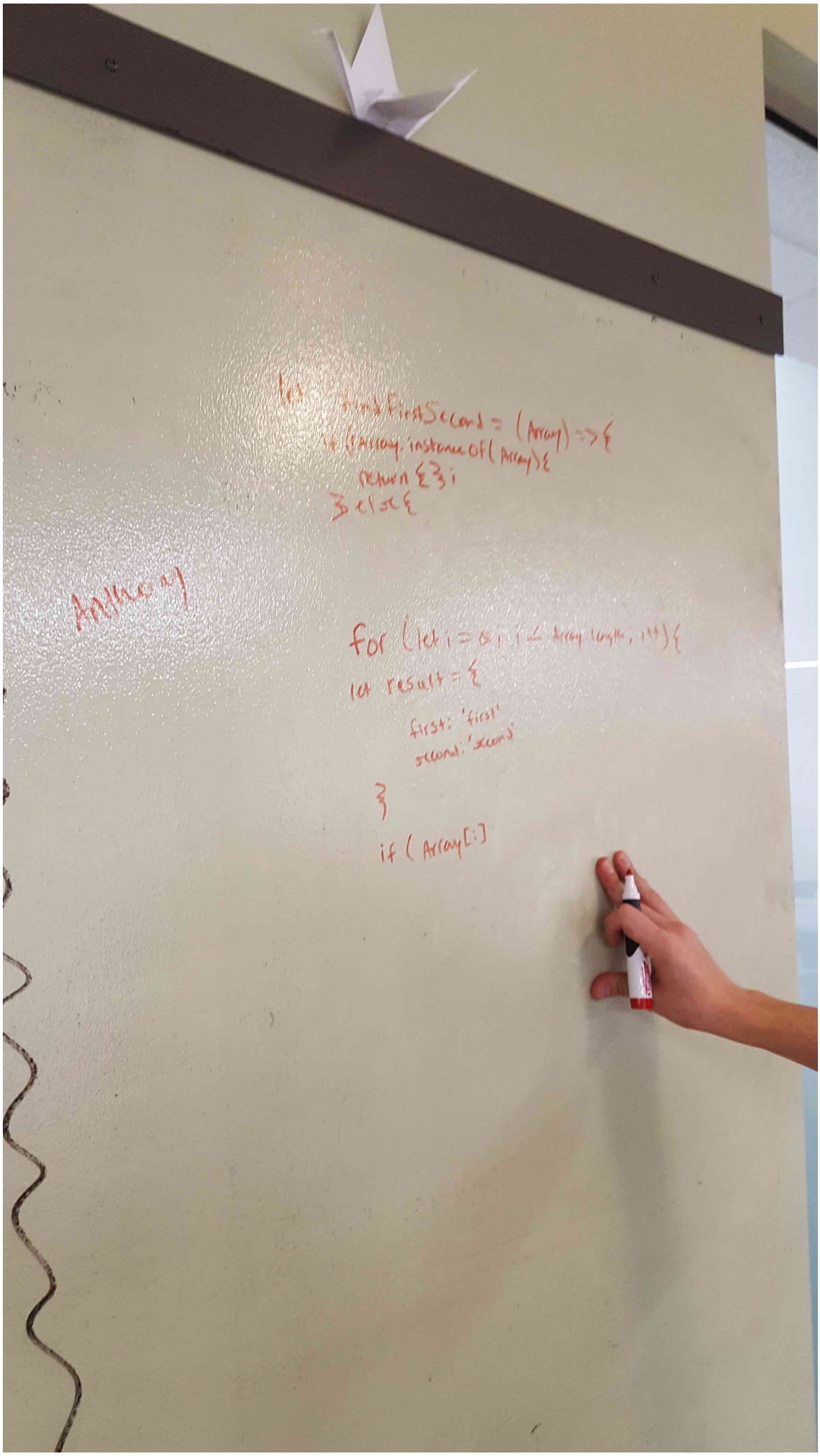
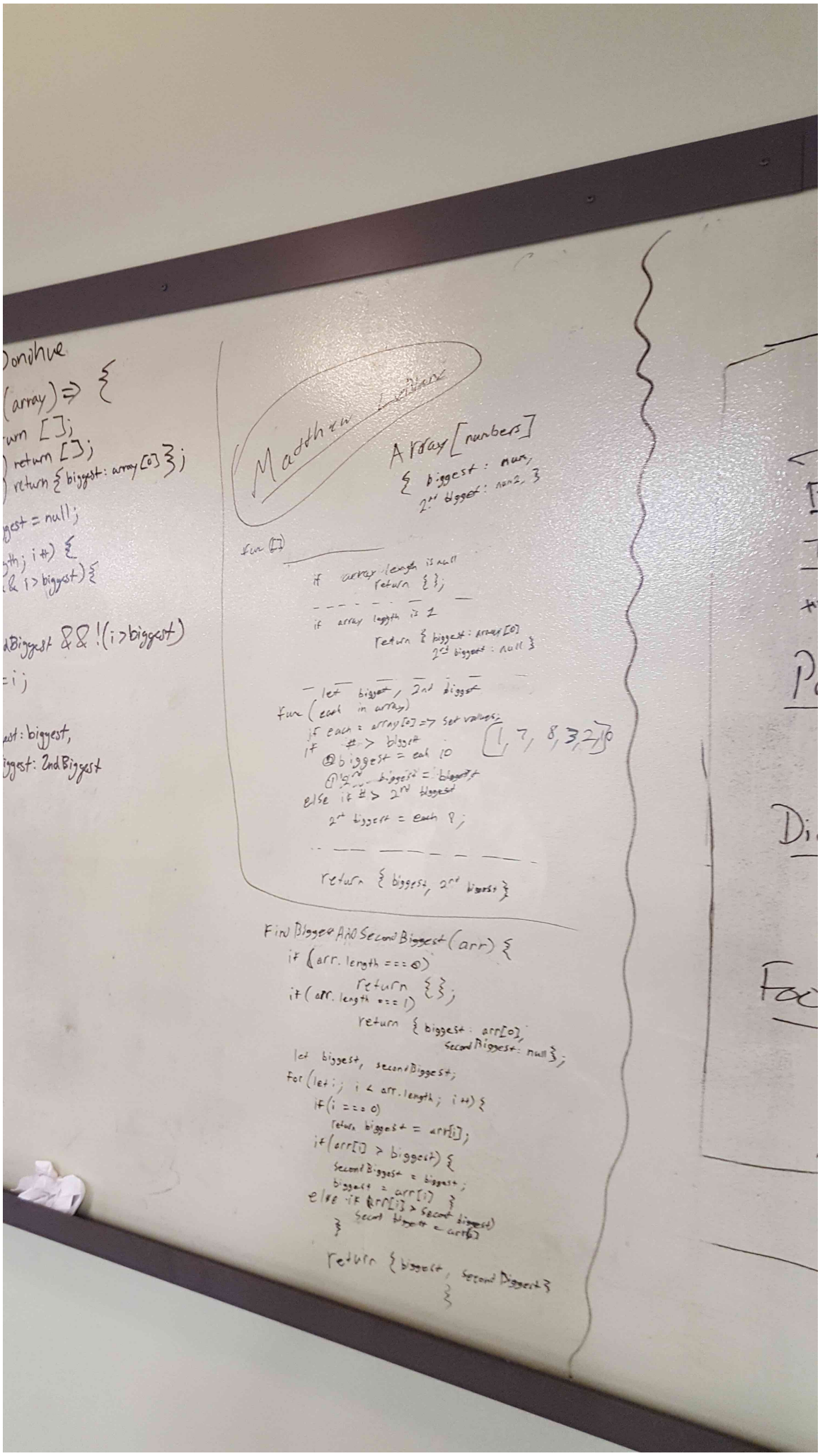

let findFirstAndSecond = (Array) => {
if (Array.instanceOf(Array){
return {};
} else {
Anything
for (let i = 0; i <= Array.length; i++){
let result = {
first: 'first'
second: 'second'
}
if (Array[:]


Array [numbers]
{ biggest : num,
2nd bigger : num2, }
if array length is null
return { };
if array length is 1
return { biggest : array[0]
2nd biggest : null }
let biggest, 2nd biggest
for (each in array)
if each = array[0] => set values;
if # > biggest
biggest = each 10
2nd biggest = biggest
else if # > 2nd biggest
2nd biggest = each 8;
return { biggest, 2nd biggest }
[1, 7, 8, 3, 2, 10]
Find Biggest And Second Biggest (arr) {
if (arr.length === 0)
return { };
if (arr.length === 1)
return { biggest : arr[0],
secondBiggest: null };
let biggest, secondBiggest;
for (let i; i < arr.length; i ++) {
if(i === 0)
return biggest = arr[i];
if(arr[i] > biggest) {
secondBiggest = biggest;
biggest = arr[i] }
else if (arr[i] > secondBiggest)
secondBiggest = arr[i]
}
return { biggest, secondBiggest }
}

tests: array = [] , array = null/undefined
array = [1], [1,2]

assumed; Array only has numbers

find biggest & 2nd biggest elements

Seth Donahue

Pseudo:

f biggestNumbers (array) => {

→ loop through array to find biggest

if (i > 2nd Biggest & i > biggest)
biggest = i
else (i > 2nd Biggest & !(i > biggest)
2nd Biggest = i

return { biggest: biggest,
2nd biggest: 2nd biggest
}

Check for edge cases

if(array.length===0) return []
if(array.length===1) retur array
if(array is not and array) return []

Code

Seth Donahue

```
let biggestNumbers = (array) => {
  if(!array.isArray()) return [];
  if(array.length===0) return [];
  if(array.length===1) return { biggest: array[0] };
  let biggest = 2nd Biggest = null;
  for (let i=0; array.length; i++) {
    if(i > 2nd Biggest && i > biggest){
      biggest = i;
    } else if(i > 2nd Biggest && !(i > biggest)
      2nd Biggest = i;

  return { biggest: biggest,
           2nd Biggest: 2nd Biggest
         };
```

tests array = [] , array = null / undefined
array = [1], [1,2]
assumed: Array only has numbers
find biggest & 2nd biggest elements

Seth Donahue

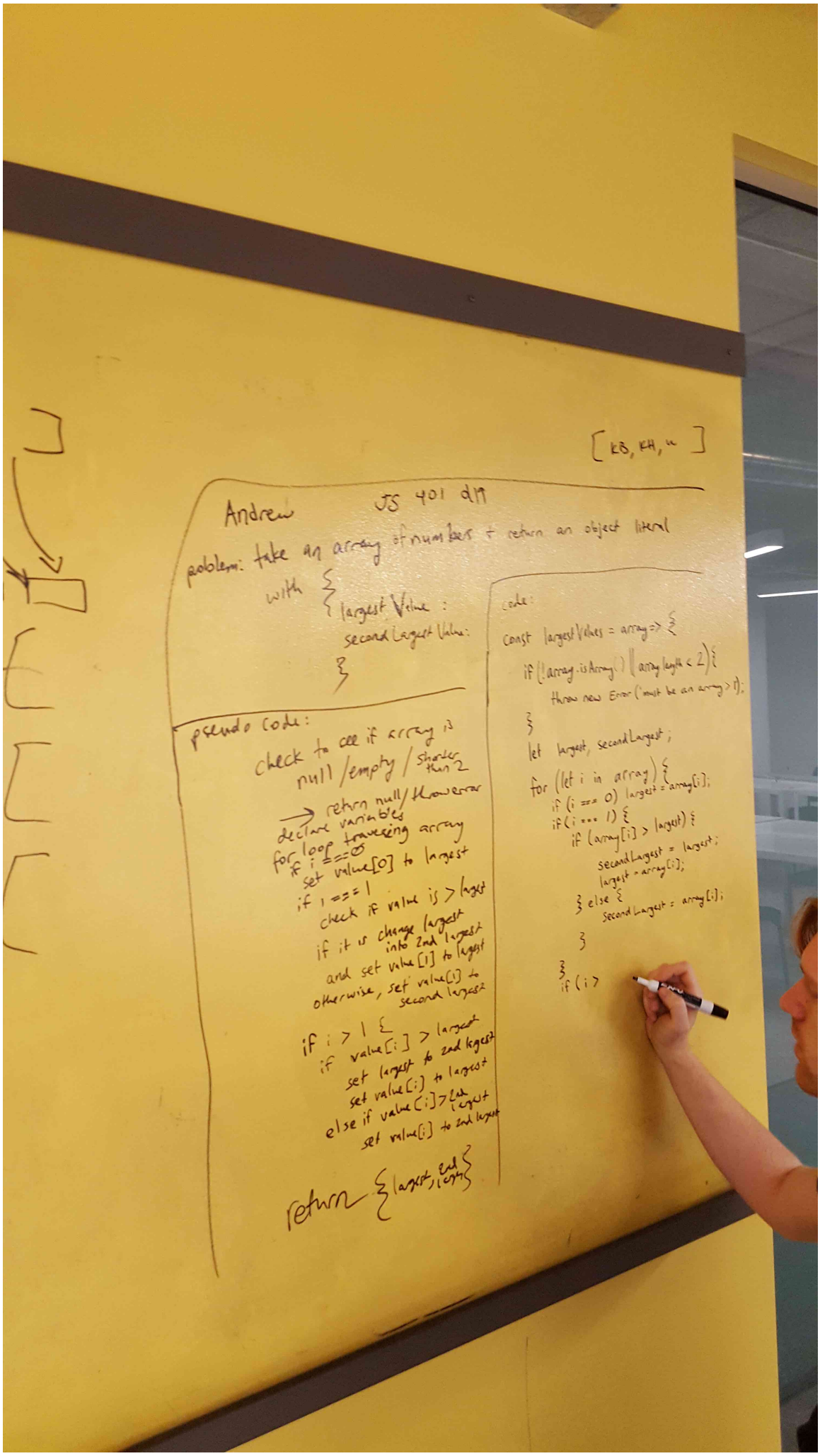
Pseudo:

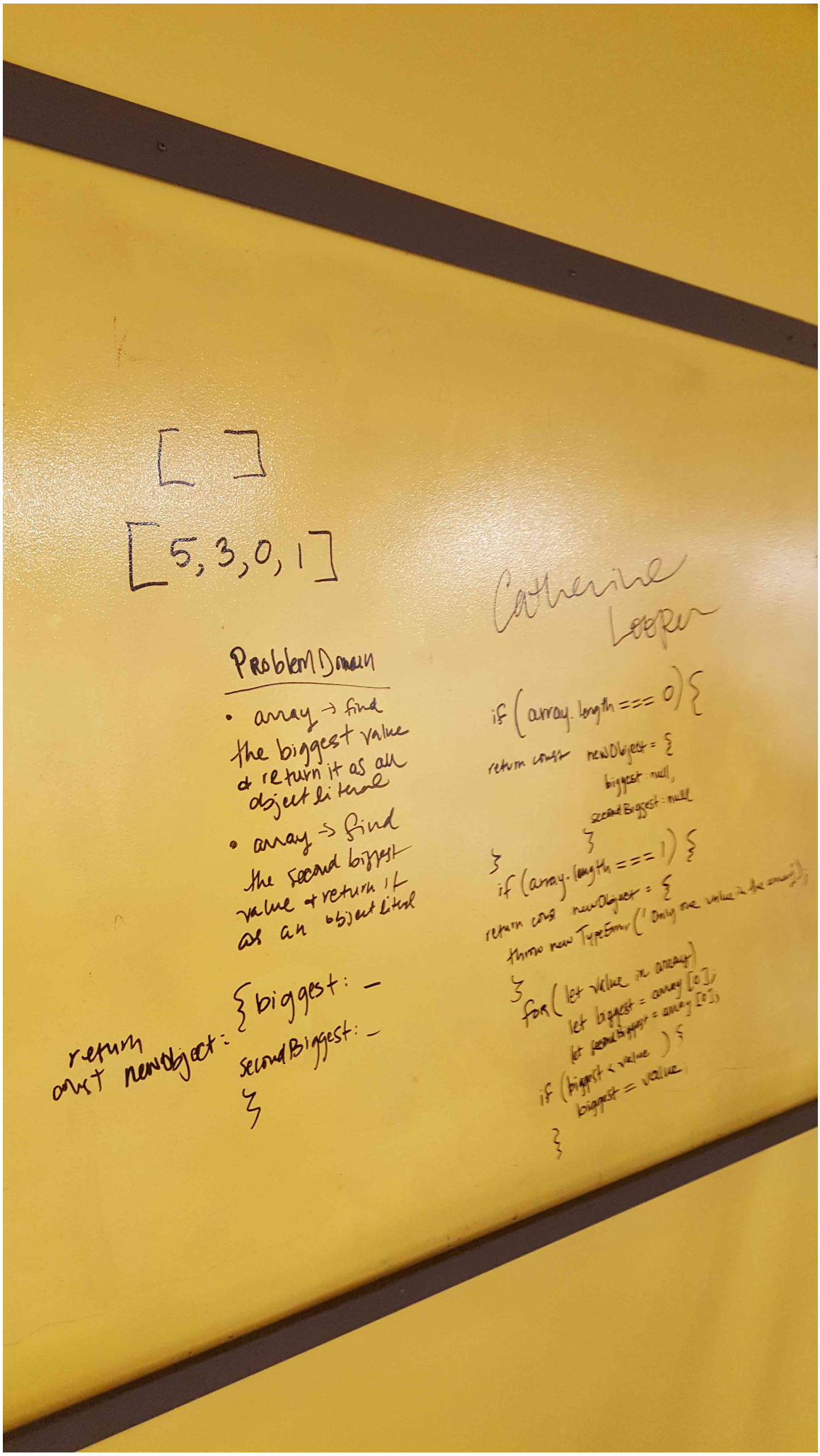
function biggestNumbers (array) => {
→ loop through array to find biggest
if (i > 2nd Biggest & i > biggest)
biggest = i
else (i > 2nd Biggest & !(i > biggest)
2nd Biggest = i
return { biggest: biggest,
2nd biggest: 2nd biggest
}

Create for edge cases
if (array.length === 0) return []
if (array.length === 1) retur array
if (array is not an array) return []

Code Seth Donahue

```
let biggestNumbers = (array) => {
  if(!array.isArray()) return [];
  if(array.length===0) return [];
  if(array.length===1) return { biggest: array[0] }
  let biggest = 2ndBiggest = null;
  for (let i=0; array.length; i++) {
    if(i>2ndBiggest && i>biggest) {
      biggest = i;
    } else if(i>2ndBiggest && !(i>biggest
      2ndBiggest = i;

  return { biggest: biggest,
           2ndBiggest: 2ndBiggest
         };
```


[KB, KH, u
Andrew JS 401 d11
problem: take an array of numbers + return an object literal
with {
largestValue :
secondLargestValue:
}
pseudo code:
check to see if array is
null / empty / shorter than 2
return null/throw error
declare variables
for loop traversing array
if i ===0
set value[0] to largest
if i === 1
check if value is > largest
if it is change largest into 2nd largest
and set value[1] to largest
otherwise, set value[i] to secondLargest
if i > 1 {
if value[i] > largest
set largest to 2nd largest
set value[i] to largest
else if value[i] > 2nd Largest
set value[i] to 2nd largest
return {largest, 2nd Largest}
code:
const largestValues = array => {
if (!array.isArray() || array.length < 2){
throw new Error('must be an array');
}
let largest, secondLargest;
for (let i in array) {
if (i === 0) largest = array[i];
if (i === 1) {
if (array[i] > largest) {
secondLargest = largest;
largest = array[i];
} else {
secondLargest = array[i];
}
}
if (i >


[ ]
[5, 3, 0, 1]
Problem Domain
• array → find the biggest value & return it as an object literal
• array → find the second biggest value & return it as an object literal
return const newObject: { biggest: –
secondBiggest: –
}
Catherine
Looper
if (array.length === 0) {
return const newObject = {
biggest: null,
secondBiggest: null
}
}
if (array.length === 1) {
return const newObject = {
throw new TypeError('Only one value in the array');
}
for (let value in array)
let biggest = array[0];
let secondBiggest = array[0];
if (biggest < value) {
biggest = value
}

1. FIND BIGGEST & SEE BIGGEST in the ARRAY
RETURN THEIR AS OBJ

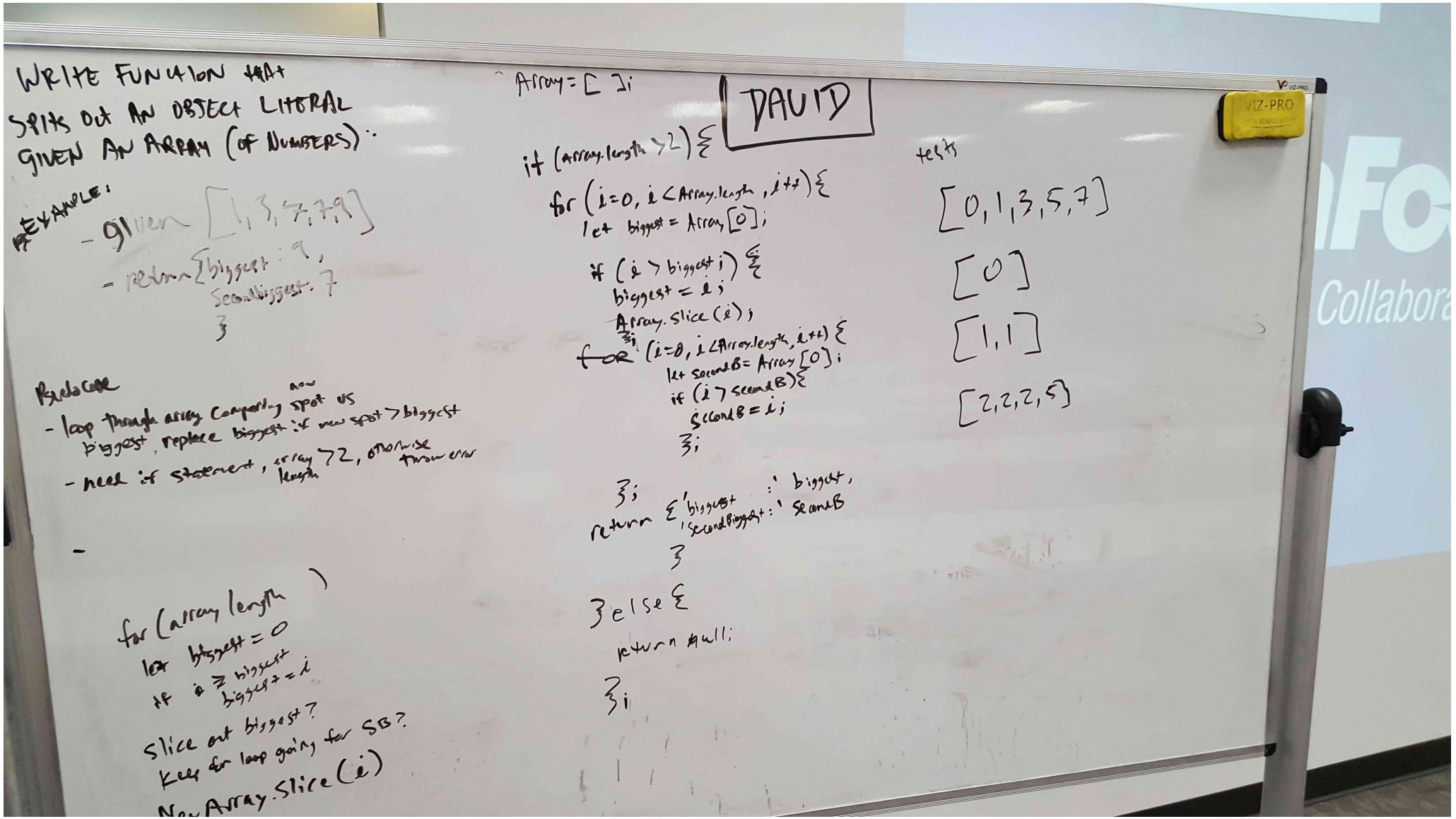
-EDGE CASES
1 EMPTY ARRAY
2. NOT AN ARRAY
3. ONE IN THE ARRAY
4. MANY TYPES in ARRAY

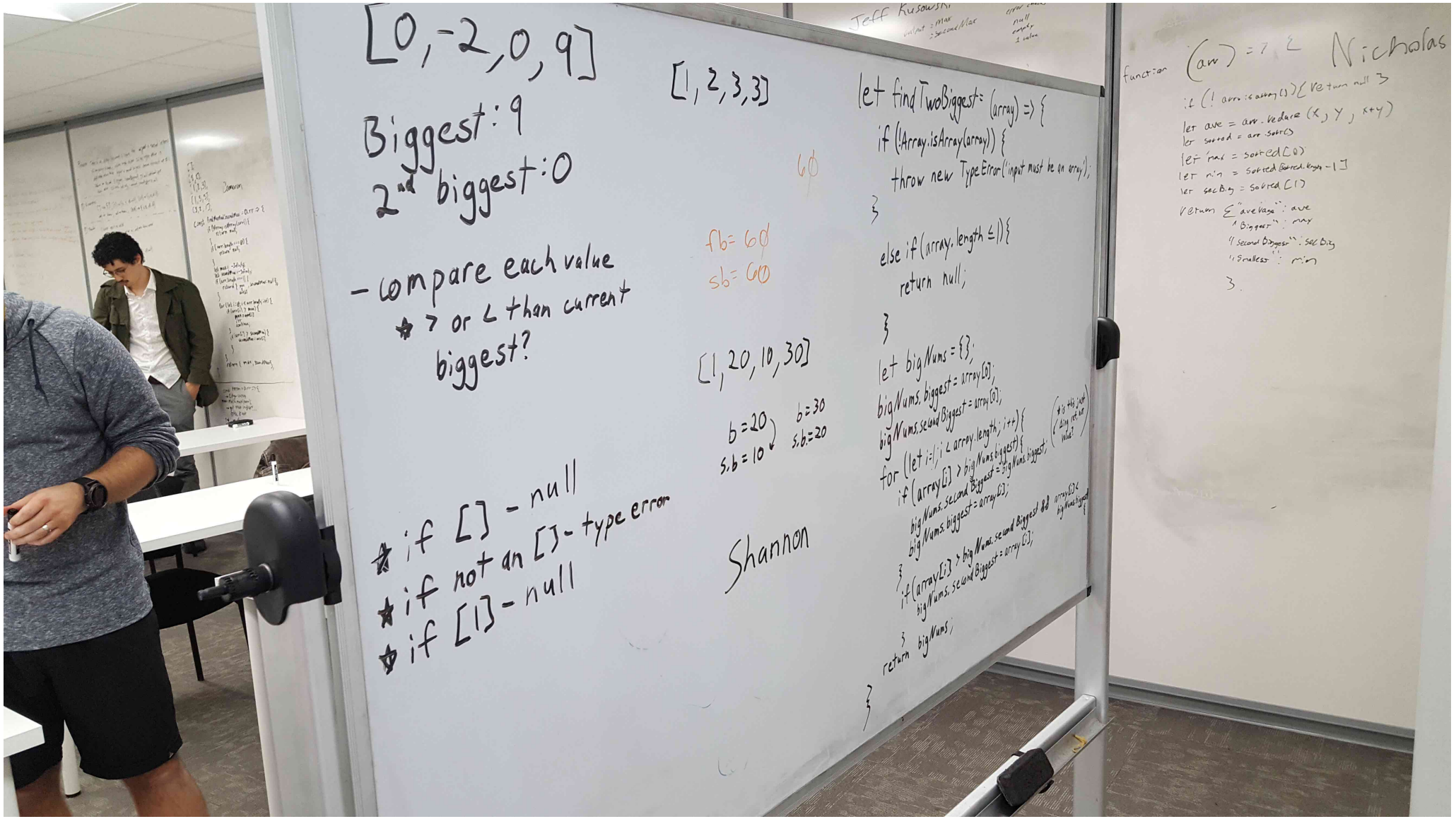
```
let bigBuffer = (array) => {
  if (!Array.isArray(i))
    RETURN {};
  if (array.length === 0)
    RETURN console.error('array is empty');
  if (array.length === 1)
    RETURN { BIG: array[0], biggest: array[0] }

  for (let value in array) {
    let biggest = array[0];
    let big = null;
    if ( value > biggest)
      biggest = array[value];
      big   = array[0];

    else {
      biggest = array[0];
      big = array[value;

    }

  RETURN { min
```

Pedro positivic

TEST
ASSING BIG and BIGGEST TO ARRAY[0]

[1,2,3]
array[0]=1 - FIRST value
BIG=1
BIGGEST=1
assign BIGGEST to ARRAY[0]
BIGGEST = 1 BIG = 1 SAME CASE
array[1]=2 value is bigger than BIGGEST
assing to BIGGEST
and assign ARRAY[0] to var BIG
BIGGEST=2
BIG = 1
array[2]=2


WRITE FUNCTION THAT
SPITS OUT AN OBJECT LITERAL
GIVEN AN ARRAY (OF NUMBERS)::
EXAMPLE:
- given [1,3,7,7,9]
- return {biggest: 9,
secondBiggest: 7
}
Pseudocode
- loop through array comparing spot vs biggest, replace biggest if new spot > biggest
- need if statement, array length > 2, otherwise throw error
-
for (array length)
let biggest = 0
if i ≥ biggest
biggest = i
slice out biggest?
Keep for loop going for SB?
New Array.slice(i)
Array = [ ];
DAVID
if (array.length > 2) {
for (i=0, i < Array.length, i++) {
let biggest = Array[0];
if (i > biggest i) {
biggest = i;
Array.slice(i);
for (i=0, i < Array.length, i++) {
let secondB = Array[0];
if (i > secondB) {
secondB = i;
};
return {'biggest' : biggest,
'secondBiggest' : secondB
}
} else {
return null;
};
tests
[0,1,3,5,7]
[0]
[1,1]
[2,2,2,5]
VIZ-PRO
Collabora


[0,-2,0,9]
Biggest: 9
2nd biggest: 0
- compare each value
* > or < than current biggest?
* if [] - null
* if not an [] - type error
* if [] - null
[1,2,3,3]
60
fb= 60
sb= 60
[1,20,10,30]
b=20
s.b=10
b=30
s.b=20
Shannon
let findTwoBiggest= (array) => {
if (!Array.isArray(array)) {
throw new TypeError('input must be an array');
}
else if (array.length <1){
return null;
}
let bigNums = {};
bigNums.biggest = array[0];
bigNums.secondBiggest = array[0];
for (let i=1; i < array.length; i++) {
if (array[i] > bigNums.biggest) {
bigNums.secondBiggest = bigNums.biggest;
bigNums.biggest = array[i];
}
if (array[i] > bigNums.secondBiggest && array[i] < bigNums.biggest) {
bigNums.secondBiggest = array[i];
}
}
return bigNums;
}
Nicholas
function (arr) => {
if (! arr.isArray()){return null}
let ave = arr.reduce (x, y, x+y)
let sorted = arr.sort()
let max = sorted[0]
let min = sorted[sorted.length-1]
let secBig = sorted[1]
return {"average": ave
"Biggest": max
"Second Biggest": secBig
"Smallest": min
}

```
function (arr) => {  Nicholas C
  if (! arr.isArray()){ return null }

  let ave = arr.reduce(x, y, x+y)
  let sorted = arr.sort()
  let max = sorted[0]
  let min = sorted[sorted.length-1]
  let secBig = sorted[1]

  return {"average": ave
          "Biggest": max
          "Second Biggest": secBig
          "Smallest": min

  }.
```

Jeff Kusowski

input = array

output = max
= second Max

error checks
null
empty
1 value

Sample [5, 10, 7, 16]

```
function (array) => {
  if (array === null || array.length === 0 || array.length === 1)
    return null;

  let biggest = array[0];
  let secondBiggest = array[0];

  for (i = 0; i < array.length; i++)
    if (array[i] > biggest) {
      secondBiggest = biggest;
      biggest = array[i];
    } else {
      if (array[i] > secondBiggest)
        secondBiggest = array[i];
    }
  }
  return {biggest, secondBiggest}
```

Assume
it's an array
all contents are numbers

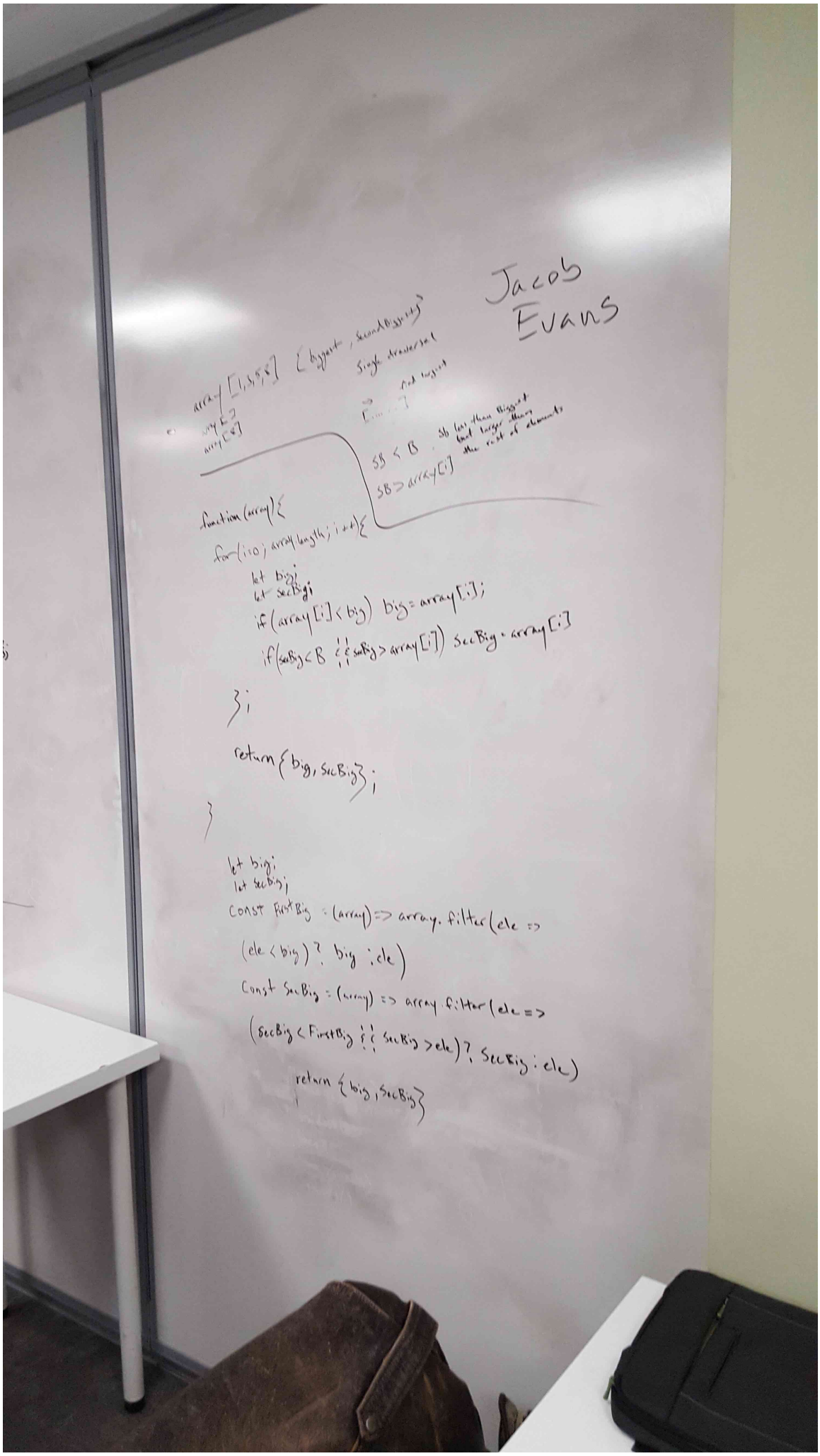
Test
biggest = 16
secondBiggest = 10

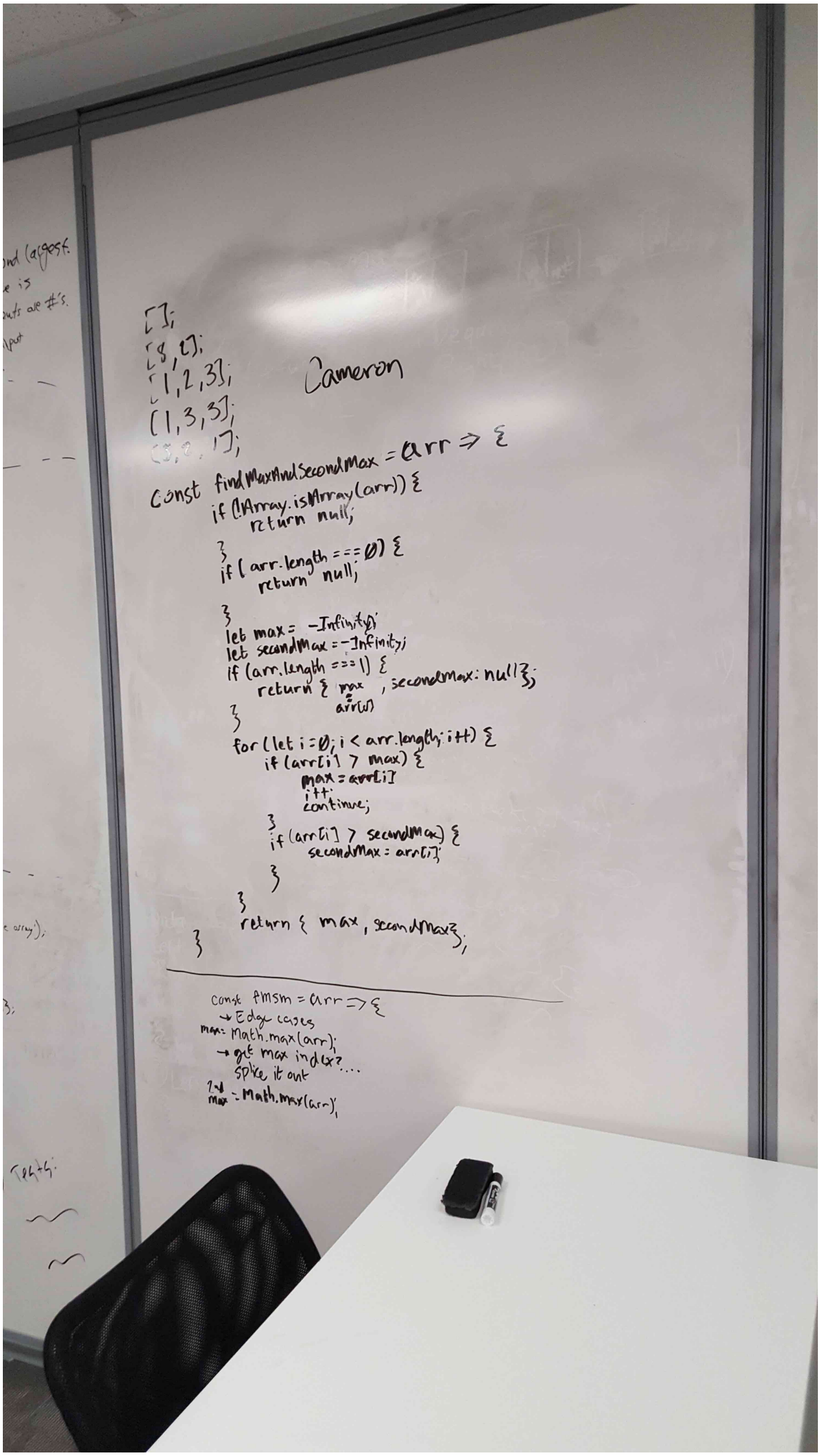
.map
.reduce for total
.sort

output biggest, secondBiggest, min, average

```
function (array) => {
  if (array === null || array.length === 0 || array.length === 1)
    return null;
  let average = (array.reduce((acc, curr) => {
    return acc + curr;
  }) / array.length

  array.sort((a, b) => { a - b })

  let biggest = array[0];
  let secondBiggest = array[1];
  let min = array[array.length];
  return {biggest, secondBiggest, average, min}
```

I forget the syntax of sort


Jacob
Evans
array = [1,3,5,8] [biggest, secondBiggest]
Single traversal
find largest
[.....]
sB less than biggest
but larger than any
the rest of elements
sB > array[i]
function (array){
for(i=0; array.length; i++){
let big;
let secBig;
if(array[i] < big) big = array[i];
if(secBig < B || secBig > array[i]) secBig = array[i]
};
return {big, secBig};
}
let big;
let secBig;
Const FirstBig = (array) => array.filter(ele =>
(ele < big) ? big : ele)
Const SecBig = (array) => array.filter(ele =>
(secBig < FirstBig || secBig > ele) ? secBig : ele)
return {big, secBig}


[];
[8, 2];
[1,2,3];
[1,3,3];
Cameron
const findMaxAndSecondMax = arr => {
if (!Array.isArray(arr)) {
return null;
}
if (arr.length === 0) {
return null;
}
let max = -Infinity;
let secondMax = -Infinity;
if (arr.length === 1) {
return { max: arr[0], secondMax: null };
}
for (let i = 0; i < arr.length; i++) {
if (arr[i] > max) {
max = arr[i]
i++;
continue;
}
if (arr[i] > secondMax) {
secondMax = arr[i];
}
}
return { max, secondMax };
}
const fmsm = arr => {
→ Edge cases
max: Math.max(arr);
→ get max index? ...
splice it out
2nd max = Math.max(arr);

Rob Reed 12/01/17 WB challenge

① Problem: Take in an array (assumed) & return the largest & second largest. If the array is empty, return empty object. If the largest value is duplicated, return it as largest & second largest. Assume elements are #'s.
Return an object {biggest, secondBiggest}. If null/undefined input throw error. If array has only 1 element, secondBiggest is null.

② Examples:
[] → {}, [4] → {b: 4, sB: null}, [2,5] → {b:5, sB:2}
und. → throw, null → throw, [8,8] → {b:8, sB:8}

③ Pseudo:
1) check input is valid
   a) null ⇒ throw error or und → throw error
2) check length
   a) array.length === 0 ⇒ {}
   b) array.length === 1 ⇒ {b: a[0], sB: null}
3) if here, data is valid & length > 1
   a) compare 1st & 2nd elements
      i) a[0] > a[1]
         i) b = a[0], sB = a[1]
      ii) else
         i) b = a[1], sB = a[0]
   b) iterate through i=2 → array.length
      i) if a[i] > b, sB = b, b = a[i]
      ii) else if a[i] > sB, sB = a[i]
   c) return {b: b, sB: sB};

④
```
const biggestTwo = array => {
  if (!array)
    throw new TypeError('input "array" must be an object of type array');
  if (array.length < 1)
    return {};
  else if (array.length < 2)
    return { biggest: array[0], secondBiggest: null};
  let biggest = array[0], secondBiggest = array[1];
  if (secondBiggest > biggest)
    [biggest, secondBiggest] = [secondBiggest, biggest];
  for (let i=2; i < array.length; i++) {
    if (array[i] > biggest) {
      [secondBiggest, biggest] = [biggest, array[i]];
    } else if (array[i] > secondBiggest)
        secondBiggest = array[i];
  }
  return { biggest: biggest,
           secondBiggest: secondBiggest
         };
```

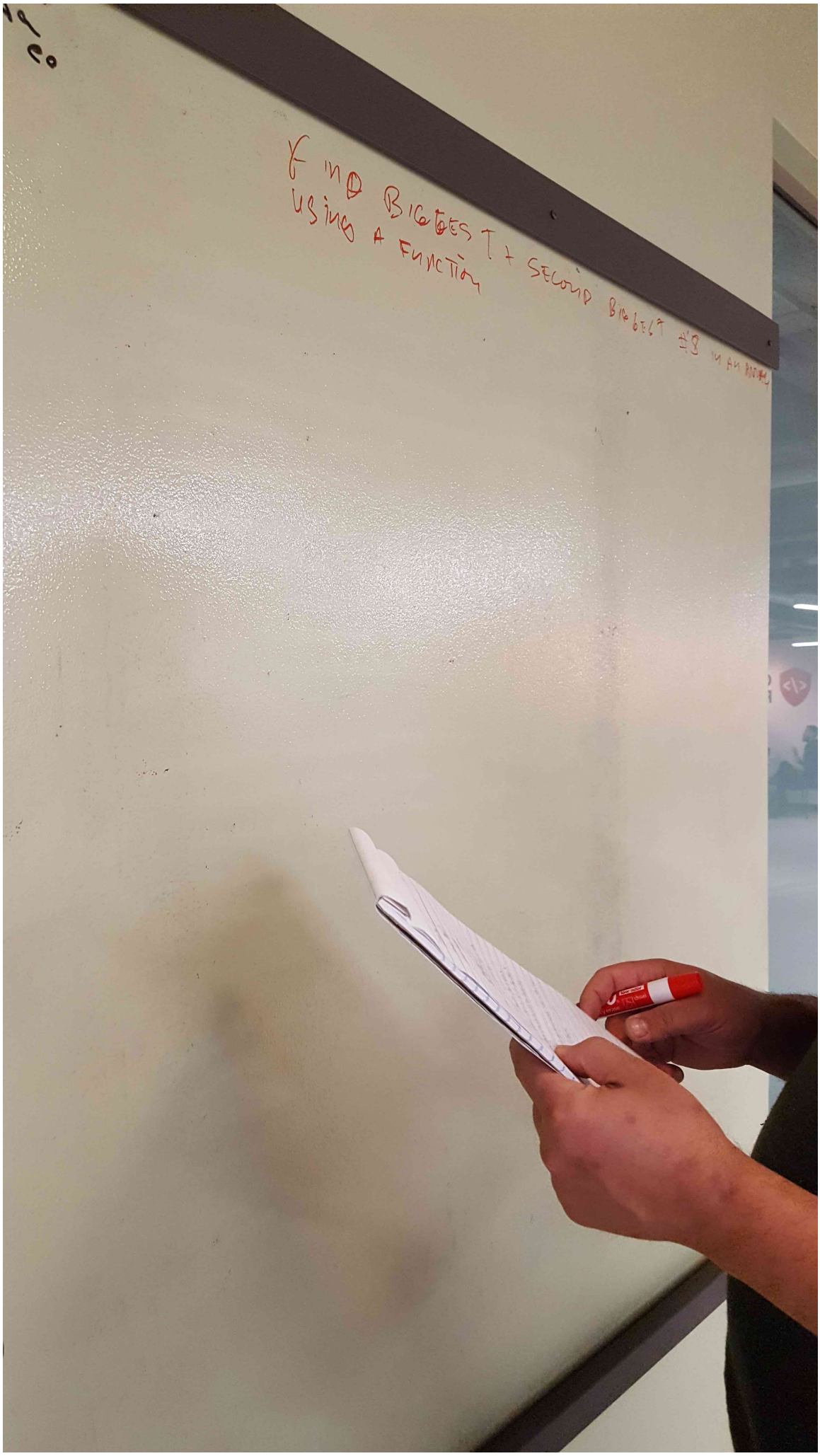
⑤ Tests:

Kerry Nordstrom

Restate: Create function which [illegible] parameter & returns object literal composed of the biggest & second biggest as key/value pairs. Do this w/ vanilla JS.

Short examples & Pseudo
- if empty array, return empty object ({ })
- if single value, return object w/ this value as both in object ({ biggest: x, secondBiggest: x })
- if expected values in array, return object with values as expected ({ biggest: x, secondBiggest: y })

```
const findNumbers = (array) => {
  let resultObj = {};
  if (array.length === 0){
    return {};
  }
  if (array.isArray()) {
    for(var i in array) {
      if(array[i] => array[i+1]){
        resultObj.biggest = array[i];
      } else if{
        (0 < array[i] < resultObj.biggest && array[i] < resultObj.biggest)
        resultObj.secondBiggest = array[i]
      } else {
        return resultObj;
      };
    };
  };
}
```


FIND BIGGEST + SECOND BIGGEST
USING A FUNCTION

? what transforms?

1) gather user input
   - ↳ in file
   - ↳ Out file
   - ↳ transform

   } index.js

2) Read in file w/ fs } reader.js

3) Parse bitmap buffer into metadata object } parser.js

4) Use metadata object to transform the file's buffer } transform.js

5) write mutated data to out file } writer.js

CREATE LIST O
ALLOWABLE CHA
[a-zA-Z0-9_]

DEFINE LENGT
{4,16}

WRAP WITH ^ $

```
const firstAndSecond = (array) => {
  var first = array[0];
  var second = null;
  var len = array.length;

  for (var i=1; i<len; i++){
    e= array[i];
    e>first ? first=e; second=i; continue:;
    second<e<first? second=e:;
  }
  return {
    'biggest Value' : first,
    'second Biggest' : second
  }
}
```

return null Here

Phelan (still testing →)

[0]
[1]
[1, 1]
[2, 1]
[1, 2]
[1, 1, 1]
[2, 2, 1]
[1, 2, 2]

[1, 2, 3]