</talentlabs>

Lecture 2
Array Algorithms Questions

</talentlabs>

Agenga

- Array Questions
- Question 1 - Missing Number
- Question 2 - Finding Duplicate Number
- Question 3 - Finding Number Pairs

Array Algorithms

</talentlabs>

Arrays

- The most commonly used type of data structure in any programming languages
- Most of the algorithm questions would involve using array in some ways
- Usually there are a couple techniques:
  - Temporary Variables
  - Looping Twice
  - Nested Loops
  - Sort First

Question 1 - Finding Missing Number

</talentlabs>

Finding Missing Number

You have an unsorted integer array (without duplicate number) and some numbers are missing in this array. In order to make the array consecutive. How do you find that missing number?

Example 1:
Input Array: [1, 3, 7] → Missing 2, 4, 5, 6

Example 2:
Input Array: [3, 7, 1] → Missing 2, 4, 5, 6

What is your solution?

What is your solution?

Hint: Confirm the range of the array first.

Example Algorithm

1. We go thru the array and find the max and min number using a for loop (Remember how we find max number last time?)
2. Use another for loop to go thru all the numbers between min and max, and check if the number exists in the input array. For those not exists, store them in a “temp result array”
3. Return the “temp result array” as output.

Assignment Question 1

Part 1 - Finding Max and Min

First, we need to have a function to find the max and min number of the array. (In order to find the range of the array)

```
arr = [3, 7, 1]
let max = arr[0];
let min = arr[0];

for(let i = 1 ;i<arr.length;i++){
    if(arr[i] > max){
        max = arr[i];
    }
    if(arr[i] < min){
        min = arr[i];
    }
}
```

We First assume the max and min are the first number in the array

Then, We use a for-loop to loop though the whole array to find the real max and min number by comparison

```
arr = [3, 7, 1]
let max = arr[0];
let min = arr[0];

for(let i = 1 ;i<arr.length;i++){
    if(arr[i] > max){
        max = arr[i];
    }
    if(arr[i] < min){
        min = arr[i];
    }
}
```

</talentlabs>

Assignment Question 1

Part 2 - Check for Missing Numbers

In the step, we first need to add a temp result array to store the missing number.

Then we will loop through the numbers from min to max, and check if we are missing some of the numbers

```
let tempResultArray = []

for (let i = min; i<= max; i++){
  if (arr.indexOf(i) === -1){
    tempResultArray.push(i)
  }
}

console.log(tempResultArray)
```

Create an array to store the final results

Check if the number exists in the array by using a for loop and "indexOf" Function.

If a number is missing, then we push it to the result array

```
let tempResultArray = []

for (let i = min; i<= max; i++){
  if (arr.indexOf(i) === -1){
    tempResultArray.push(i)
  }
}

console.log(tempResultArray)
```

Techniques that we used

1. Temporary Variables
2. Looping Twice

Question 2 - Finding Duplicate Number

</talentlabs>

Finding Missing Number

You have an unsorted integer array (with duplicate number).In order to find these duplicate number, what will you do?

Example 1:
Input Array: [2, 3, 3, 7, 9, 2] → Duplicating Numbers: 2, 3

Example 2:
Input Array: [2, 3, 3, 3, 7, 9, 2] → Duplicating Numbers: 2, 3

What is your solution?

What is your solution?

Hint: Would it be easier if you sort the array first?

Example Algorithm

1. Sort the array first
2. Create a temporary result array
3. Use a for loop to go through each number in the sorted array. Compare each number the the number next to it.
4. If the next number is same as the number, then add it to the temporary result array.
5. Return the temporary array as result.

Part 1 - Sort the Array

First, we need to sort the array so the same numbers will be next to each other.

```
arr = [2, 3, 3, 7, 9, 2]
sortedArr = arr.sort()
```

We first sort the array and save it in a new variable.

Assignment Question 2

Part 2 - Check for Duplicating Numbers

In the step, we are going to leverage a for loop to check on the numbers one by one. If the number is same as the next number, then it is duplicated.

After identified the duplicating number, then we push it to the tempResultArray.

```
let tempResultArray = []

for (let i = 0; i < sortedArr.length - 1; i++){
  if (sortedArr[i] === sortedArr[i+1]){
    tempResultArray.push(sortedArr[i])
  }
}

console.log(tempResultArray)
```

We don’t need to check for the final number

Create an array to store the final results

If the number equals to the next number, that means it’s duplicated.

If a number is missing, then we push it to the result array

Input Array: [2, 2, 3, 3, 3, 7, 9]

```
let tempResultArray = []

for (let i = 0; i < sortedArr.length - 1; i++){
  if (sortedArr[i] === sortedArr[i+1]){
    tempResultArray.push(sortedArr[i])
  }
}

console.log(tempResultArray)
```

</talentlabs>

Part 3 - Avoid Duplicating Results

If we just push the number to result array without checking if it is already in the result array, then our result would have duplicating numbers too.

e.g. Input Array: [2, 2, 3, 3, 3, 7, 9] → [2, 3, 3]

```
let tempResultArray = []

for (let i = 0; i < sortedArr.length - 1; i++){
  if (sortedArr[i] === sortedArr[i+1]){
    if (tempResultArray.indexOf(sortedArr[i]) === -1){
      tempResultArray.push(sortedArr[i])
    }
  }
}

console.log(tempResultArray)
```

Additional check before pushing the number to the result array

Input Array: [2, 2, 3, 3, 3, 7, 9]

```
let tempResultArray = []

for (let i = 0; i < sortedArr.length - 1; i++){
  if (sortedArr[i] === sortedArr[i+1]){
    if (tempResultArray.indexOf(sortedArr[i]) === -1){
      tempResultArray.push(sortedArr[i])
    }
  }
}

console.log(tempResultArray)
```

</talentlabs>

Techniques that we used

1. Sorting First
2. Temporary Variable

Question 3 - Finding Number Pairs

</talentlabs>

Finding Missing Number

How to find all pairs of integer in an array whose sum is equal to a given number?

Example 1:
Input Array: [2, 3, 7, 9, 2], Target Sum: 10
Results: 3, 7

Example 2:
Input Array: [3, 4, 6, 1], Target Sum: 10
Results: 4, 6

What is your solution?

What is your solution?

Hint: Using 2 loops, nested loops

Example Algorithm

1. Use a for loop to loop through the input array
2. For each element in the input array, calculate the sum of this element with each of the other elements
3. For those that add up to our target sum, print the pair

Assignment Question 3

Part 1 - Loop Thru the Array

First, we need to loop through the array

```
let arr = [2, 3, 7, 9, 2]
let targetSum = 10

for (let i = 0; i < arr.length; i++){
  for (let j = i;  j < arr.length; j++){
    if (arr[i] + arr[j] === targetSum){
      console.log(arr[i] + ", " + arr[j])
    }
  }
}
```

We first loop thru the array elements one by one

Assignment Question 3

Part 2 - Loop Thru in a Nested Loop

Then we need to create a nested loop to go thru each of the elements again.

```
let arr = [2, 3, 7, 9]
let targetSum = 10

for (let i = 0; i < arr.length; i++){
  for (let j = 0;  j < arr.length; j++){
    if (arr[i] + arr[j] === targetSum){
      console.log(arr[i] + ", " + arr[j])
    }
  }
}
```

Create a second loop, a nested loop to loop thru the array again, so we can calculate the sum of the elements

Add up each pair of elements and see if that add up to the target sum

	2	**3**	**7**	**9**	**2**
2	4	5	9	11	4
3	5	6	10	12	5
7	9	10	14	16	9
9	11	12	16	18	11
2	4	5	9	11	4

Input Array: [2, 3, 7, 9]

```
let arr = [2, 3, 7, 9, 2]
let targetSum = 10

for (let i = 0; i < arr.length; i++){
  for (let j = 0;  j < arr.length; j++){
    if (arr[i] + arr[j] === targetSum){
      console.log(arr[i] + ", " + arr[j])
    }
  }
}
```

	2	3	7	9	2
2					
3					
7					
9					
2					

Assignment Question 3

Part 3 - Avoiding Duplicate Results

You might noticed that, the results are duplicated (i.e. both 3,7 and 7,3 are printed out). We need to update the

```
let arr = [2, 3, 7, 9]
let targetSum = 10

for (let i = 0; i < arr.length; i++){
  for (let j = i + 1;  j < arr.length; j++){
    if (arr[i] + arr[j] === targetSum){
      console.log(arr[i] + ", " + arr[j])
    }
  }
}
```

We can only calculate half of the iteration, to avoid duplicate results

	2	3	7	9	2
2	4	5	9	11	4
3	5	6	10	12	5
7	9	10	14	16	9
9	11	12	16	18	11
2	4	5	9	11	4

Techniques that we used

1. Nested Loops