

코딩 테스트 개요 및 문제 풀이를 위한 JavaScript 문법

6) JavaScript 배열 문제 풀이

코딩 테스트 알아보기 | JavaScript 핵심 문법 알아보기

강사 나동빈

코딩 테스트 개요 및 문제 풀이를 위한 JavaScript 문법

6) JavaScript 배열 문제 풀이

코딩 테스트 개요
JavaScript 배열 문제

혼자 힘으로 풀어보기

Chapter 1.
JavaScript
배열 문제

문제 제목: 최소, 최대

문제 난이도: ★☆☆☆☆

문제 유형: 기초 문법

추천 풀이 시간: 10분

코딩 테스트 개요

JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.

JavaScript 배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 최솟값을 찾는 문제다.
- 최댓값(max value)과 최솟값(min value) 정보를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

코딩 테스트 개요

JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.

JavaScript 배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 최솟값을 찾는 문제다.
- 최댓값(max value)과 최솟값(min value) 정보를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값(max value): 71

최솟값(min value): 12

코딩 테스트 개요
JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.
JavaScript
배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 최솟값을 찾는 문제다.
- 최댓값(max value)과 최솟값(min value) 정보를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값(max value): 71

최솟값(min value): 59

코딩 테스트 개요

JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.

JavaScript 배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 최솟값을 찾는 문제다.
- 최댓값(max value)과 최솟값(min value) 정보를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값(max value): 82

최솟값(min value): 59

코딩 테스트 개요

JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.

JavaScript 배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 최솟값을 찾는 문제다.
- 최댓값(max value)과 최솟값(min value) 정보를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값(max value): 82

최솟값(min value): 40

코딩 테스트 개요

JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.

JavaScript 배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 최솟값을 찾는 문제다.
- 최댓값(max value)과 최솟값(min value) 정보를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값(max value): 82

최솟값(min value): 33

코딩 테스트 개요

JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.

JavaScript 배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 최솟값을 찾는 문제다.
- 최댓값(max value)과 최솟값(min value) 정보를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값(max value): 82

최솟값(min value): 12

코딩 테스트 개요
JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.
JavaScript
배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 최솟값을 찾는 문제다.
- 최댓값(max value)과 최솟값(min value) 정보를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값(max value): 82

최솟값(min value): 12

코딩 테스트 개요

JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.

JavaScript 배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 최솟값을 찾는 문제다.
- 최댓값(max value)과 최솟값(min value) 정보를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값(max value): 82

최솟값(min value): 12

- 원소를 차례대로 하나씩 확인한다는 점에서 시간 복잡도 $O(N)$ 로 해결할 수 있다.

코딩 테스트 개요 JavaScript 배열 문제

정답 코드 예시 1)

Chapter 1. JavaScript 배열 문제

```
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

let n = Number(input[0]);
let arr = input[1].split(' ').map(Number);

/*
모든 정수는 -1,000,000보다 크거나 같고,
1,000,000보다 작거나 같은 정수이다.
*/
let minValue = 1000001; // 일단 큰 수로 초기화
let maxValue = -1000001; // 일단 작은 수로 초기화
for (let i = 0; i < n; i++) {
    if (minValue > arr[i]) minValue = arr[i];
    if (maxValue < arr[i]) maxValue = arr[i];
}
console.log(minValue, maxValue);
```

코딩 테스트 개요
JavaScript 배열 문제

정답 코드 예시 2)

Chapter 1.
JavaScript
배열 문제

```
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

let n = Number(input[0]);
let data = input[1].split(' ').map(x => Number(x));

let minValue = data.reduce((a, b) => Math.min(a, b));
let maxValue = data.reduce((a, b) => Math.max(a, b));

console.log(minValue + " " + maxValue);
```

코딩 테스트 개요
JavaScript 배열 문제

혼자 힘으로 풀어보기

Chapter 1.
JavaScript
배열 문제

문제 제목: 최댓값

문제 난이도: ★☆☆☆☆

문제 유형: 기초 문법

추천 풀이 시간: 10분

코딩 테스트 개요

JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.

JavaScript 배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 그 인덱스를 찾는 문제다.
- 계속해서 가장 큰 값을 가지는 원소의 인덱스(index)를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

코딩 테스트 개요 JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1. JavaScript 배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 그 인덱스를 찾는 문제다.
- 계속해서 가장 큰 값을 가지는 원소의 인덱스(index)를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값 인덱스(max value index): 0

코딩 테스트 개요

JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.

JavaScript 배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 그 인덱스를 찾는 문제다.
- 계속해서 가장 큰 값을 가지는 원소의 인덱스(index)를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값 인덱스(max value index): 0

코딩 테스트 개요

JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.

JavaScript 배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 그 인덱스를 찾는 문제다.
- 계속해서 가장 큰 값을 가지는 원소의 인덱스(index)를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값 인덱스(max value index): 2

코딩 테스트 개요 JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1. JavaScript 배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 그 인덱스를 찾는 문제다.
- 계속해서 가장 큰 값을 가지는 원소의 인덱스(index)를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값 인덱스(max value index): 2

코딩 테스트 개요

JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.

JavaScript 배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 그 인덱스를 찾는 문제다.
- 계속해서 가장 큰 값을 가지는 원소의 인덱스(index)를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값 인덱스(max value index): 2

코딩 테스트 개요 JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1. JavaScript 배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 그 인덱스를 찾는 문제다.
- 계속해서 가장 큰 값을 가지는 원소의 인덱스(index)를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값 인덱스(max value index): 2

코딩 테스트 개요 JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1. JavaScript 배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 그 인덱스를 찾는 문제다.
- 계속해서 가장 큰 값을 가지는 원소의 인덱스(index)를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값 인덱스(max value index): 2

코딩 테스트 개요

JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.

JavaScript 배열 문제

- 배열의 원소를 하나씩 확인하여, 최댓값과 그 인덱스를 찾는 문제다.
- 계속해서 가장 큰 값을 가지는 원소의 인덱스(index)를 업데이트한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값 인덱스(max value index): 2

- 원소를 차례대로 하나씩 확인한다는 점에서 시간 복잡도 $O(N)$ 로 해결할 수 있다.

코딩 테스트 개요 JavaScript 배열 문제

정답 코드 예시 1)

Chapter 1. JavaScript 배열 문제

```
// readline 모듈보다는 fs를 이용해 파일 전체를 읽어 들여 처리하기
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

let maxIndex = 0;
let maxValue = 0;
for (let i = 0; i < 9; i++) { // 모든 데이터를 하나씩 확인하며
    let data = Number(input[i]);
    if (maxValue < data) {
        maxValue = data;
        maxIndex = i;
    }
}

console.log(maxValue);
console.log(maxIndex + 1);
```

코딩 테스트 개요 JavaScript 배열 문제

정답 코드 예시 2)

Chapter 1. JavaScript 배열 문제

```
// readline 모듈보다는 fs를 이용해 파일 전체를 읽어 들여 처리하기
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

let data = input.map(x => Number(x));
let max = Math.max(...data)

console.log(max)
console.log(input.indexOf(max) + 1)
```

코딩 테스트 개요
JavaScript 배열 문제

혼자 힘으로 풀어보기

Chapter 1.
JavaScript
배열 문제

문제 제목: 나머지

문제 난이도: ★☆☆☆☆

문제 유형: 기초 문법

추천 풀이 시간: 10분

코딩 테스트 개요

JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.

JavaScript 배열 문제

- 각 원소를 하나씩 확인하며, 42로 나눈 나머지를 차례대로 출력한다.
- 원소를 하나씩 확인한다는 점에서 시간 복잡도는 $O(N)$ 이다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	105	88	92	215	77	193	302

코딩 테스트 개요
JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.
JavaScript
배열 문제

- 각 원소를 하나씩 확인하며, 42로 나눈 나머지를 차례대로 출력한다.
- 원소를 하나씩 확인한다는 점에서 시간 복잡도는 $O(N)$ 이다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	105	88	92	215	77	193	302

현재 출력할 값: $105 \% 42 = 21$

나머지 값들: [21]

집합: {21}

코딩 테스트 개요
JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.
JavaScript
배열 문제

- 각 원소를 하나씩 확인하며, 42로 나눈 나머지를 차례대로 출력한다.
- 원소를 하나씩 확인한다는 점에서 시간 복잡도는 $O(N)$ 이다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	105	88	92	215	77	193	302

현재 출력할 값: $88 \% 42 = 4$

나머지 값들: [21, 4]

집합: {21, 4}

코딩 테스트 개요
JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.
JavaScript
배열 문제

- 각 원소를 하나씩 확인하며, 42로 나눈 나머지를 차례대로 출력한다.
- 원소를 하나씩 확인한다는 점에서 시간 복잡도는 $O(N)$ 이다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	105	88	92	215	77	193	302

현재 출력할 값: $92 \% 42 = 8$

나머지 값들: [21, 4, 8]

집합: {21, 4, 8}

코딩 테스트 개요
JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.
JavaScript
배열 문제

- 각 원소를 하나씩 확인하며, 42로 나눈 나머지를 차례대로 출력한다.
- 원소를 하나씩 확인한다는 점에서 시간 복잡도는 $O(N)$ 이다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	105	88	92	215	77	193	302

현재 출력할 값: $215 \% 42 = 5$

나머지 값들: [21, 4, 8, 5]

집합: {21, 4, 8, 5}

코딩 테스트 개요
JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.
JavaScript
배열 문제

- 각 원소를 하나씩 확인하며, 42로 나눈 나머지를 차례대로 출력한다.
- 원소를 하나씩 확인한다는 점에서 시간 복잡도는 $O(N)$ 이다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	105	88	92	215	77	193	302

현재 출력할 값: $77 \% 42 = 35$

나머지 값들: [21, 4, 8, 5, 35]

집합: {21, 4, 8, 5, 35}

코딩 테스트 개요 JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1. JavaScript 배열 문제

- 각 원소를 하나씩 확인하며, 42로 나눈 나머지를 차례대로 출력한다.
- 원소를 하나씩 확인한다는 점에서 시간 복잡도는 $O(N)$ 이다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	105	88	92	215	77	193	302

현재 출력할 값: $193 \% 42 = 25$

나머지 값들: [21, 4, 8, 5, 35, 25]

집합: {21, 4, 8, 5, 35, 25}

코딩 테스트 개요
JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.
JavaScript
배열 문제

- 각 원소를 하나씩 확인하며, 42로 나눈 나머지를 차례대로 출력한다.
- 원소를 하나씩 확인한다는 점에서 시간 복잡도는 $O(N)$ 이다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	105	88	92	215	77	193	302

현재 출력할 값: $302 \% 42 = 8$

나머지 값들: [21, 4, 8, 5, 35, 25, 8]

집합: {21, 4, 8, 5, 35, 25}

코딩 테스트 개요 JavaScript 배열 문제

정답 코드 예시

Chapter 1. JavaScript 배열 문제

```
// readline 모듈보다는 fs를 이용해 파일 전체를 읽어 들여 처리하기
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

let data = input.map(Number);
let mySet = new Set(); // 집합 객체 생성

// 원소를 하나씩 확인하며
for (let i = 0; i < 10; i++) {
    mySet.add(data[i] % 42); // 42로 나눈 나머지를 집합의 원소로 삽입
}

// 집합에 포함된 원소의 개수 출력
console.log(mySet.size);
```

코딩 테스트 개요
JavaScript 배열 문제

혼자 힘으로 풀어보기

Chapter 1.
JavaScript
배열 문제

문제 제목: 평균은 넘겠지

문제 난이도: ★☆☆☆☆

문제 유형: 기초 문법

추천 풀이 시간: 10분

코딩 테스트 개요 JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1. JavaScript 배열 문제

[풀이 방법]

1. 먼저 N 개의 원소에 대한 평균 값을 계산한다. $O(N)$
 2. 다시 N 개의 원소를 하나씩 확인하여 평균 이상인지 확인한다. $O(N)$
- 전체 시간 복잡도는 $O(N)$ 이다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

코딩 테스트 개요 JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1. JavaScript 배열 문제

[풀이 방법]

1. 먼저 N 개의 원소에 대한 평균 값을 계산한다. $O(N)$
 2. 다시 N 개의 원소를 하나씩 확인하여 평균 이상인지 확인한다. $O(N)$
- 전체 시간 복잡도는 $O(N)$ 이다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

계산된 평균 값: 52.143

평균 이상인 값들: [71, 59, 82, 68]

따라서, 평균을 넘는 학생의 비율은 $4/7 = 57.143\%$ 이다.

코딩 테스트 개요

JavaScript 배열 문제

정답 코드 예시

Chapter 1.

JavaScript 배열 문제

```
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

let testCases = Number(input[0]);
for (let t = 1; t <= testCases; t++) { // 각각의 테스트 케이스를 확인
  let data = input[t].split(' ').map(Number);
  let n = data[0]; // 첫 번째 수는 데이터의 개수
  let summary = 0; // n개의 점수에 대하여 평균 계산
  for (let i = 1; i <= n; i++) {
    summary += data[i];
  }
  let average = summary / n;
  let cnt = 0; // 점수가 평균을 넘는 학생 수 계산
  for (let i = 1; i <= n; i++)
    if (data[i] > average) cnt += 1;
  // 점수가 평균을 넘는 학생의 비율을 소수점 아래 셋째 자리까지 출력
  console.log(`${(cnt / n * 100).toFixed(3)}%`);
}
```


코딩 테스트 개요
JavaScript 배열 문제

혼자 힘으로 풀어보기

Chapter 1.
JavaScript
배열 문제

문제 제목: 평균

문제 난이도: ★☆☆☆☆

문제 유형: 기초 문법

추천 풀이 시간: 10분

코딩 테스트 개요
JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.
JavaScript
배열 문제

- 다음의 과정을 통해 시간 복잡도 $O(N)$ 으로 문제를 해결할 수 있다.
 1. 가장 먼저, 최댓값을 계산한다.
 2. 이후에 각 값을 하나씩 확인하여 새로운 값 $X/M \times 100$ 을 계산하여 평균을 구한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

코딩 테스트 개요
JavaScript 배열 문제

문제 풀이 핵심 아이디어

Chapter 1.
JavaScript
배열 문제

- 다음의 과정을 통해 시간 복잡도 $O(N)$ 으로 문제를 해결할 수 있다.
 1. 가장 먼저, 최댓값을 계산한다.
 2. 이후에 각 값을 하나씩 확인하여 새로운 값 $X/M \times 100$ 을 계산하여 평균을 구한다.

인덱스(index)	0	1	2	3	4	5	6
값(value)	71	59	82	40	33	12	68

최댓값(M): 82

코딩 테스트 개요 JavaScript 배열 문제

정답 코드 예시

Chapter 1. JavaScript 배열 문제

```
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

let n = Number(input[0]);
let scores = input[1].split(' ').map(Number);

let maxValue = scores.reduce((a, b) => Math.max(a, b));
let updated = [];
for (let i = 0; i < n; i++) { // 수정된 원소 하나씩 저장
    updated.push(scores[i] / maxValue * 100);
}

// 배열에 포함된 원소의 평균 출력
console.log(updated.reduce((a, b) => a + b) / n);
```