

JavaScript 정렬 알고리즘

6) 정렬 문제 풀이 ①

정렬 문제 풀이 | 알고리즘의 기본이 되는 정렬 알고리즘 이해하기

강사 나동빈

JavaScript

정렬 알고리즘

6) 정렬 문제 풀이 ①

JavaScript 정렬
정렬 문제 풀이

혼자 힘으로 풀어보기

JavaScript
정렬
정렬 문제 풀이

문제 제목: 세수정렬

문제 난이도: ★☆☆☆☆

문제 유형: 정렬

추천 풀이 시간: 10분

JavaScript 정렬
정렬 문제 풀이

문제 해결 아이디어

JavaScript
정렬
정렬 문제 풀이

- 이 문제는 JavaScript에서 제공하는 `sort()` 메서드를 이용해 해결할 수 있다.
- JavaScript의 경우 원하는 기준에 따라서 객체에 대한 정렬을 수행하는 기능을 제공한다.
- 단순히 세 개의 수가 주어지므로, 이를 배열에 담아 **정렬(sorting)을 수행**한다.

JavaScript 정렬 정렬 문제 풀이

정답 코드 예시 1)

JavaScript 정렬 정렬 문제 풀이

```
// fs 모듈을 이용해 파일 전체를 읽어와 문자열로 저장하기
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

let arr = input[0].split(' ').map(Number);
// 오름차순 정렬 수행
arr.sort(function(a, b) {
  return a - b;
});

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

JavaScript 정렬 정렬 문제 풀이

정답 코드 예시 2)

JavaScript 정렬 정렬 문제 풀이

```
function selectionSort(arr) { // 선택 정렬 함수
  for (let i = 0; i < arr.length; i++) {
    let minIndex = i; // 가장 작은 원소의 인덱스
    for (let j = i + 1; j < arr.length; j++)
      if (arr[minIndex] > arr[j]) minIndex = j;
    let temp = arr[i]; // 스와프(swap)
    arr[i] = arr[minIndex];
    arr[minIndex] = temp;
  }
}

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

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

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

JavaScript 정렬
정렬 문제 풀이

혼자 힘으로 풀어보기

JavaScript
정렬
정렬 문제 풀이

문제 제목: 수 정렬하기

문제 난이도: ★☆☆☆☆

문제 유형: 정렬

추천 풀이 시간: 10분

JavaScript 정렬
정렬 문제 풀이

문제 해결 아이디어

JavaScript
정렬
정렬 문제 풀이

- 이 문제는 JavaScript에서 제공하는 `sort()` 메서드를 이용해 해결할 수 있다.
- 단순히 N 개의 수를 입력 받아, 이를 오름차순 정렬한 결과를 반환할 수 있다.
- N 의 값이 최대 1,000이므로, 시간 복잡도 $O(N^2)$ 의 알고리즘을 사용할 수도 있다.

JavaScript 정렬 정렬 문제 풀이

정답 코드 예시 1)

JavaScript 정렬 정렬 문제 풀이

```
// fs 모듈을 이용해 파일 전체를 읽어와 문자열로 저장하기
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

let n = Number(input[0]);
let arr = [];
for (let i = 1; i <= n; i++) { // 한 줄씩 입력받아 배열에 넣기
    arr.push(Number(input[i]));
}
arr.sort(function(a, b) { // 오름차순 정렬 수행
    return a - b;
})

let answer = "";
for (let i = 0; i < arr.length; i++) {
    answer += arr[i] + "\n";
}
console.log(answer);
```

JavaScript 정렬 정렬 문제 풀이

정답 코드 예시 2)

JavaScript 정렬 정렬 문제 풀이

```
function selectionSort(arr) { // 선택 정렬 함수
  for (let i = 0; i < arr.length; i++) {
    let minIndex = i; // 가장 작은 원소의 인덱스
    for (let j = i + 1; j < arr.length; j++)
      if (arr[minIndex] > arr[j]) minIndex = j;
    let temp = arr[i]; // 스와프(swap)
    arr[i] = arr[minIndex];
    arr[minIndex] = temp;
  }
}

let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');
let n = Number(input[0]);
let arr = [];
for (let i = 1; i <= n; i++) {
  arr.push(Number(input[i]));
}
selectionSort(arr);
let answer = "";
for (let i = 0; i < arr.length; i++) {
  answer += arr[i] + "\n";
}
console.log(answer);
```

JavaScript 정렬
정렬 문제 풀이

혼자 힘으로 풀어보기

JavaScript
정렬
정렬 문제 풀이

문제 제목: 수 정렬하기 2

문제 난이도: ★☆☆☆☆

문제 유형: 정렬

추천 풀이 시간: 15분

- 이 문제는 JavaScript에서 제공하는 `sort()` 메서드를 이용해 해결할 수 있다.
- JavaScript의 경우 원하는 기준에 따라서 객체에 대한 정렬을 수행하는 기능을 제공한다.
- JavaScript에서 제공하는 `sort()`는 시간 복잡도 $O(N \log N)$ 을 보장한다.
- 따라서, 본 문제에서는 N 의 최대 크기가 100만이라는 점에서 `sort()`로 문제를 해결할 수 있다.
- 본 문제는 시간 복잡도 $O(N^2)$ 의 정렬 알고리즘으로는 **시간 초과**를 받는다.

JavaScript 정렬 정렬 문제 풀이

정답 코드 예시

JavaScript 정렬 정렬 문제 풀이

```
// fs 모듈을 이용해 파일 전체를 읽어와 문자열로 저장하기
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

let n = Number(input[0]);
let arr = [];
for (let i = 1; i <= n; i++) {
    arr.push(Number(input[i]));
}
// 오름차순 정렬 수행
arr.sort(function(a, b) {
    return a - b;
});

let answer = "";
for (let i = 0; i < arr.length; i++) {
    answer += arr[i] + "\n";
}
console.log(answer);
```

JavaScript 정렬
정렬 문제 풀이

혼자 힘으로 풀어보기

JavaScript
정렬
정렬 문제 풀이

문제 제목: K번째 수

문제 난이도: ★☆☆☆☆

문제 유형: 정렬

추천 풀이 시간: 10분

JavaScript 정렬 정렬 문제 풀이

문제 해결 아이디어

JavaScript 정렬 정렬 문제 풀이

- 이 문제는 JavaScript에서 제공하는 `sort()` 메서드를 이용해 해결할 수 있다.
- JavaScript의 경우 원하는 기준에 따라서 객체에 대한 정렬을 수행하는 기능을 제공한다.
- JavaScript에서 제공하는 `sort()`는 시간 복잡도 $O(N \log N)$ 을 보장한다.
- 따라서, 본 문제에서는 N 의 최대 크기가 500만이라는 점에서 `sort()`로 문제를 해결할 수 있다.
- 본 문제는 시간 복잡도 $O(N^2)$ 의 정렬 알고리즘으로는 **시간 초과**를 받는다.

JavaScript 정렬 정렬 문제 풀이

정답 코드 예시

JavaScript 정렬 정렬 문제 풀이

```
// fs 모듈을 이용해 파일 전체를 읽어와 문자열로 저장하기
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

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

// 오름차순 정렬 수행
arr.sort(function(a, b) {
    return a - b;
});

// 앞에서부터 K번째 수를 출력
console.log(arr[k - 1]);
```