

JavaScript DFS 알고리즘 DFS 문제 풀이

DFS 문제 풀이 | 코딩 테스트에서 자주 등장하는 DFS 알고리즘 이해하기

강사 나동빈

JavaScript

DFS 알고리즘

DFS 문제 풀이

JavaScript DFS
DFS 문제 풀이

혼자 힘으로 풀어보기

JavaScript
DFS
DFS 문제 풀이

문제 제목: 차이를 최대로

문제 난이도: ★★☆☆☆

문제 유형: 깊이 우선 탐색, 백트래킹

추천 풀이 시간: 40분

JavaScript DFS

DFS 문제 풀이

문제 해결 아이디어

JavaScript

DFS

DFS 문제 풀이

- N 개의 정수로 이루어진 배열 A 가 주어진다.
- 배열에 들어있는 정수의 순서를 적절히 바꿔서 다음 식의 최댓값을 구하는 프로그램을 작성한다.

$$|A[0] - A[1]| + |A[1] - A[2]| + \cdots + |A[N - 2] - A[N - 1]|$$

JavaScript DFS DFS 문제 풀이

정답 코드 예시

JavaScript DFS DFS 문제 풀이

```
function dfs(depth) {
  if (depth == n) { // 각 순열에 대한 처리
    let current = 0; // 공식 계산
    for (let i = 0; i < n - 1; i++) current += Math.abs(result[i] - result[i + 1]);
    maxValue = Math.max(maxValue, current);
  }
  for (let i = 0; i < n; i++) { // 각 수를 N개 선택하는 모든 순열 계산
    if (visited[i]) continue; // 이미 고른 수라면 무시하도록
    visited[i] = true; // 현재 선택한 수 방문 처리
    result.push(arr[i]);
    dfs(depth + 1); // 재귀 함수 호출
    visited[i] = false; // 현재 선택한 수 방문 처리 취소
    result.pop();
  }
}

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

let n = Number(input[0]); // 정수의 개수(N)
let arr = input[1].split(' ').map(Number); // 전체 정수 배열 입력
let visited = new Array(10).fill(false); // 방문 처리 배열
let result = [];
let maxValue = -1e9;
dfs(0);
console.log(maxValue);
```

JavaScript DFS
DFS 문제 풀이

혼자 힘으로 풀어보기

JavaScript
DFS
DFS 문제 풀이

문제 제목: 연산자 끼워넣기

문제 난이도: ★★☆☆☆

문제 유형: 깊이 우선 탐색, 완전 탐색, 백트래킹

추천 풀이 시간: 50분

JavaScript DFS
DFS 문제 풀이

문제 해결 아이디어

JavaScript
DFS
DFS 문제 풀이

- N 개의 수로 이루어진 수열이 있다.
- 또한, 수 사이에 삽입할 수 있는 $N - 1$ 개의 연산자가 주어진다.

JavaScript DFS

DFS 문제 풀이

문제 해결 아이디어

JavaScript

DFS

DFS 문제 풀이

- N 개의 수가 주어졌을 때(N 은 최대 11), 각 수 사이에 사칙연산 중 하나를 삽입한다.
- 사용 가능한 총 연산자의 수는 $N - 1$ 개이다.
- 따라서 단순히 모든 경우를 고려해도 시간 초과를 받지 않는다.
- 따라서 연산자를 끼워 넣는 모든 경우의 수를 고려하여 값을 계산한다.
- 완전 탐색을 위해 DFS를 사용할 수 있다.

- 연산자 예시: 덧셈(+) 1개, 뺄셈(-) 1개, 곱셈(X) 1개

7

5

6

3

- 모든 **경우의 수(6가지)**는 다음과 같다.

1. $7 + 5 - 6 \times 3$

2. $7 + 5 \times 6 - 3$

3. $7 - 5 + 6 \times 3$

4) $7 - 5 \times 6 + 3$

5) $7 \times 5 + 6 - 3$

6) $7 \times 5 - 6 + 3$

JavaScript DFS

DFS 문제 풀이

정답 코드 예시

JavaScript DFS

DFS 문제 풀이

```
function dfs(index, value) {
  if (index == n) {
    minValue = Math.min(minValue, value);
    maxValue = Math.max(maxValue, value);
    return;
  }
  if (add > 0) {
    add--;
    dfs(index + 1, value + arr[index]);
    add++;
  }
  if (sub > 0) {
    sub--;
    dfs(index + 1, value - arr[index]);
    sub++;
  }
  if (mul > 0) {
    mul--;
    dfs(index + 1, value * arr[index]);
    mul++;
  }
  if (div > 0) {
    div--;
    dfs(index + 1, ~~(value / arr[index])); // 나눌 때는 나머지를 제거 (C++14과 동일)
    div++;
  }
}
```

JavaScript DFS

DFS 문제 풀이

정답 코드 예시

JavaScript

DFS

DFS 문제 풀이

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

let n = Number(input[0]); // 수의 개수
let arr = input[1].split(' ').map(Number); // 연산을 수행하고자 하는 수 리스트
// 덧셈(+), 뺄셈(-), 곱셈(*), 나눗셈(/)의 연산자 개수
let [add, sub, mul, div] = input[2].split(' ').map(Number);

// 최솟값과 최대값 초기화
let minValue = 1e9;
let maxValue = -1e9;

// DFS 메서드 호출
dfs(1, arr[0]);

// 최대값과 최솟값을 차례대로 출력
console.log(maxValue);
console.log(minValue);
```