

JavaScript 누적합

누적합 알고리즘 문제 풀이

누적합 알고리즘 문제 풀이 | 코딩 테스트에서 자주 등장하는 누적합 이해하기

강사 나동빈

JavaScript

누적합

누적합 알고리즘 문제 풀이

JavaScript 누적합
누적합 문제 풀이

혼자 힘으로 풀어보기

JavaScript
누적합
누적합
문제 풀이

문제 제목: 합 구하기

문제 난이도: ★★☆☆☆

문제 유형: 누적합

추천 풀이 시간: 40분

JavaScript 누적합
누적합 문제 풀이

문제 풀이 핵심 아이디어

JavaScript
누적합
누적합
문제 풀이

- N 개의 수가 담긴 배열이 주어진다.
- M 개의 구간 (i, j) 가 주어졌을 때 i 번째 수부터 j 번째 수까지의 구간 합을 계산한다.

JavaScript 누적합 누적합 문제 풀이

정답 코드 예시

JavaScript 누적합

누적합
문제 풀이

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

let n = Number(input[0]);
let data = input[1].split(' ').map(Number);
let m = Number(input[2]);

let prefixSum = [0]; // 누적 합(prefix sum)
let summary = 0;
for (let i of data) {
    summary += i;
    prefixSum.push(summary);
}

let answer = '';
for (let query = 3; query <= m + 2; query++) { // 각 쿼리(query) 처리
    let [i, j] = input[query].split(' ').map(Number);
    answer += (prefixSum[j] - prefixSum[i - 1]) + '\n';
}
console.log(answer);
```

JavaScript 누적합
누적합 문제 풀이

혼자 힘으로 풀어보기

JavaScript
누적합
누적합
문제 풀이

문제 제목: 나머지 합

문제 난이도: ★★☆☆☆

문제 유형: 누적합

추천 풀이 시간: 60분

JavaScript 누적합
누적합 문제 풀이

문제 풀이 핵심 아이디어

JavaScript
누적합
누적합
문제 풀이

[일반적인 구간 합 문제]

- 기본적으로 배열에서 각 위치까지의 **누적 합(prefix sum)**을 이용한다.
- 배열의 구간 합은 $sum[end] - sum[start - 1]$ 으로 $O(1)$ 의 시간으로 계산할 수 있다.

JavaScript 누적합
누적합 문제 풀이

문제 풀이 핵심 아이디어

JavaScript
누적합
누적합
문제 풀이

[예시]

- 인덱스 0은 사용하지 않는다.
- $arr = [0, 3, 2, 5, 6, 7]$
- $sum = [0, 3, 5, 10, 16, 23]$
- 1번째부터 4번째까지 구간 합: $sum[4] - sum[0] = 16$
- 3번째부터 5번째까지 구간 합: $sum[5] - sum[2] = 18$

[목표]

- 연속 구간의 합이 M 의 배수인 경우의 수를 구한다.
→ $(start, end)$ 조합에 대해 $(sum[end] - sum[start - 1]) \% M == 0$ 인 개수를 구해야 한다.
- 모듈로($\%$) 연산은 분배 법칙이 성립한다.
- $sum[end] \% M - sum[start - 1] \% M == 0$ 인 개수를 구하는 것과 같다.
- 결과적으로 $sum[end] \% M == sum[start - 1] \% M$ 인 개수를 구하는 문제로 치환된다.

JavaScript 누적합
누적합 문제 풀이

문제 풀이 핵심 아이디어

JavaScript
누적합
누적합
문제 풀이

- 실제로 구간 합 연산을 수행할 때 사용하는 것을 고려하면 $end \neq start - 1$ 이다.
→ 구간의 크기가 1인 경우인 $end = start$ 인 상황이라고 해도 문제 없다.

[문제 해결 방법]

- 각 위치까지의 누적 합에 대하여 모듈로(%) 연산을 이용해 M으로 나눈 나머지를 저장한다.

JavaScript 누적합 누적합 문제 풀이

문제 풀이 핵심 아이디어

JavaScript
누적합
누적합
문제 풀이

[문제 해결 방법]

- 각 위치까지의 누적 합에 대하여 모듈로(%) 연산을 이용해 M으로 나눈 나머지를 저장한다.
- 문제의 요구사항은 *processed* 배열에서 서로 다른 인덱스 (x, y) 에 대하여
- $processed[x] == processed[y]$ 인 경우의 수를 세는 것과 같다.
- 따라서, *processed* 배열에 존재하는 각 값에 대하여 카운트를 세준다.

[예시] 특정한 값을 가지는 카운트가 5라고 하면, 5에서 2개를 고르는 조합의 수를 고려하면 된다.

→ $5C2 = 10$ 이다.

JavaScript 누적합 누적합 문제 풀이

문제 풀이 핵심 아이디어

JavaScript
누적합
누적합
문제 풀이

[예시 1]

- $N = 10, M = 50$ 이고, 다음과 같다고 가정하자.
- $arr = [10, 4, 7, 8, 6, 9, 13, 12, 21, 24]$
- 따라서, 누적 합은 다음과 같다.
- $sum = [10, 14, 21, 29, 35, 44, 57, 69, 90, 114]$
- 따라서, 누적 합을 M 으로 나눈 나머지는 다음과 같다.
- $processed = [0, 4, 1, 4, 0, 4, 2, 4, 0, 4]$

[예시 1] $arr = [10, 4, 7, 8, 6, 9, 13, 12, 21, 24]$

- $sum = [10, 14, 21, 29, 35, 44, 57, 69, 90, 114]$
- $processed = [0, 4, 1, 4, 0, 4, 2, 4, 0, 4]$
- 결과적으로 $counter$ 변수는 다음과 같다.

구간 합에서 $start = 0$ 인 경우 고려

인덱스	0	1	2	3	4
카운터	$3 + 1$	1	1	0	5

- 예를 들어 $counter[4]$ 를 보면, 값이 5번 등장했다.
- 이 경우, $5C2 = 10$ 개의 경우의 수가 존재하는 것이다.

JavaScript 누적합 누적합 문제 풀이

정답 코드 예시

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

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

let sum = [0]; // 누적 합(prefix sum)
for (let i = 1; i <= n; i++) {
    sum[i] = sum[i - 1] + arr[i];
}
```

JavaScript
누적합
누적합
문제 풀이

JavaScript 누적합 누적합 문제 풀이

정답 코드 예시

JavaScript
누적합
누적합
문제 풀이

```
let processed = []; // 누적 합(prefix sum)을 M으로 나눈 나머지
let counter = {}; // 각 나머지 값에 대한 개수를 저장하고 있는 카운터
for (let i = 0; i <= n; i++) {
    processed[i] = sum[i] % m;
    if (processed[i] in counter) counter[processed[i]] += 1;
    else counter[processed[i]] = 1;
}

let result = 0;
for (let i = 0; i < m; i++) {
    // counter[i]개에서 2개를 고르는 조합
    if (i in counter) result += parseInt(counter[i] * (counter[i] - 1) / 2);
}
console.log(result);
```