고급 알고리즘

구간 합 빠르게 계산하기

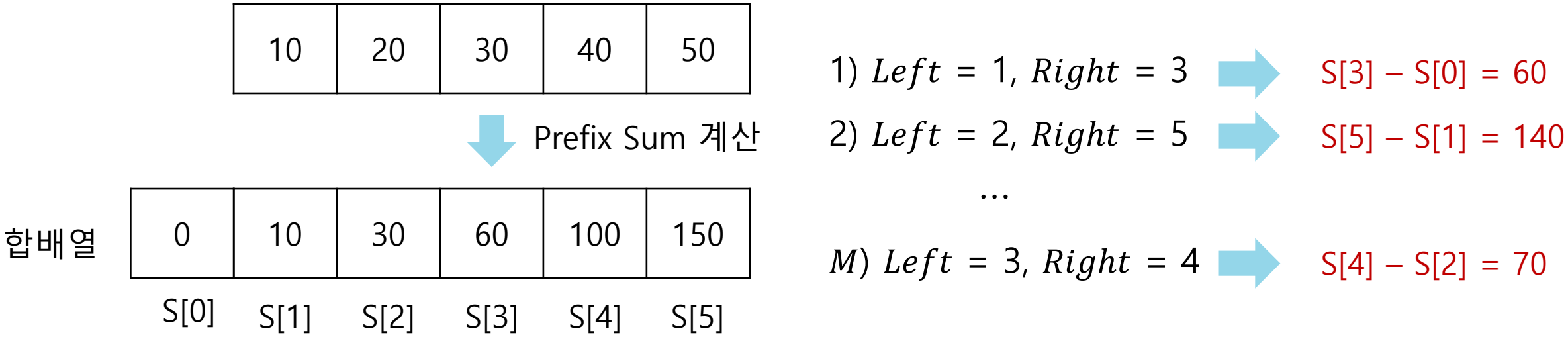
- 구간 합 문제: 연속적으로 나열된 N개의 수가 있을 때 특정 구간의 모든 수를 합한 값을 계산하는 문제
- 구간 합은 합 배열을 이용하여 시간 복잡도를 더 줄이기 위해 사용하는 특수한 목적의 알고리즘
- 예를 들어 5개의 데이터로 구성된 수열 {10, 20, 30, 40, 50}이 있다고 가정합시다.
  - 두 번째 수부터 네 번째 수까지의 합은 20 + 30 + 40 = 90입니다.

구간 합 빠르게 계산하기: 문제 설명

- N개의 정수로 구성된 수열이 있습니다.
- M개의 쿼리(Query) 정보가 주어집니다.
  - 각 쿼리는 $Left$와 $Right$으로 구성됩니다.
  - 각 쿼리에 대하여 $[Left, Right]$ 구간에 포함된 데이터들의 합을 출력해야 합니다.
- 수행 시간 제한은 $\boldsymbol{O(N+M)}$입니다.

구간 합 빠르게 계산하기: 문제 해결 아이디어

- 접두사 합(Prefix Sum): 배열의 맨 앞부터 특정 위치까지의 합을 미리 구해 놓은 것
- 접두사 합을 활용한 알고리즘은 다음과 같습니다.
  - N개의 수 위치 각각에 대하여 접두사 합을 계산하여 P에 저장합니다.
  - 매 M개의 쿼리 정보를 확인할 때 구간 합은 $P[Right] - P[Left - 1]$입니다.



구간 합 빠르게 계산하기: 코드 예시 (Python)

```python
# 데이터의 개수 N과 데이터 입력받기
n = 5
data = [10, 20, 30, 40, 50]

# 접두사 합(Prefix Sum) 배열 계산
sum_value = 0
prefix_sum = [0]
for i in data:
    sum_value += i
    prefix_sum.append(sum_value)

# 구간 합 계산(세 번째 수부터 네 번째 수까지)
left = 3
right = 4
print(prefix_sum[right] - prefix_sum[left - 1])
```

실행 결과

```
70
```

구간 합 빠르게 계산하기: 코드 예시 (C++)

```cpp
#include <bits/stdc++.h>

using namespace std;

int n = 5; // 데이터의 개수 N과 데이터 입력받기
int arr[] = {10, 20, 30, 40, 50};
int prefixSum[6];

int main() {
    // 접두사 합(Prefix Sum) 배열 계산
    int sumValue = 0;

    for (int i = 0; i < n; i++) {
        sumValue += arr[i];
        prefixSum[i + 1] = sumValue;
    }

    // 구간 합 계산(세 번째 수부터 네 번째 수까지)
    int left = 3;
    int right = 4;
    cout << prefixSum[right] - prefixSum[left - 1] << '\n';
}
```

실행 결과

```
70
```

구간 합 빠르게 계산하기: 코드 예시 (Java)

```java
import java.util.*;

class Main {
    public static int n = 5; // 데이터의 개수 N과 데이터 입력받기
    public static int arr[] = {10, 20, 30, 40, 50};
    public static int[] prefixSum = new int[6];

    public static void main(String[] args) {
        // 접두사 합(Prefix Sum) 배열 계산
        int sumValue = 0;

        for (int i = 0; i < n; i++) {
            sumValue += arr[i];
            prefixSum[i + 1] = sumValue;
        }

        // 구간 합 계산(세 번째 수부터 네 번째 수까지)
        int left = 3;
        int right = 4;
        System.out.println(prefixSum[right] - prefixSum[left - 1]);
    }
}
```

실행 결과

```
70
```

구간 합

핵심

구간 합 구하기

시간 제한 0.5초 | 난이도 실버III | 백준 온라인 저지 11659번

수 N개가 주어졌을 때 i 번째 수에서 j 번째 수까지의 합을 구하는 프로그램을 작성하시오.

입력

1번째 줄에 수의 개수 N(1 ≤ N ≤ 100,000), 합을 구해야 하는 횟수 M(1 ≤ M ≤ 100,000), 2번째 줄에 N개의 수가 주어진다. 각 수는 1,000보다 작거나 같은 자연수다. 3번째 줄부터는 M개의 줄에 합을 구해야 하는 구간 i와 j가 주어진다.

출력

총 M개의 줄에 입력으로 주어진 i 번째 수에서 j 번째 수까지의 합을 출력한다.

구간 합

예제 입력 1

```
5 3              // 데이터의 개수, 질의 개수
5 4 3 2 1        // 구간 합을 구할 대상 배열
1 3
2 4
5 5
```

예제 출력 1

```
12
9
1
```

구간 합

01단계 문제 분석하기

1. 문제에서 수의 개수와, 합을 구해야 하는 횟수는 최대 100,000이다.
2. 구간마다 합을 매번 계산하면 0.5초 안에 모든 구간 합 계산을 끝낼 수 없다.
3. 구간 합을 이용한다.

구간 합

02단계 손으로 풀어보기

1 N개의 수를 입력받음과 동시에 합 배열을 생성합니다.

합 배열 공식

```
S[i] = S[i-1] + A[i]
```

인덱스	1	2	3	4	5
배열 A	5	4	3	2	1
합 배열 S	5	9	12	14	15

구간 합

02단계 손으로 풀어보기

2 구간 i ~ j가 주어지면 구간 합을 구하는 공식으로 정답을 출력합니다.

구간 합 공식

```
S[j] - S[i-1]
```

질의1(1, 3): S[3] - S[0] = 12 0 = 12

질의2(2, 4): S[4] S[1] = 14 5 = 9

질의3(5, 5): S[5] S[4] = 15 14 = 1

구간 합(Java예시)

04단계 코드 구현하기

```java
import java.io.*;
import java.util.*;

class GFG {
    public static void main (String[] args) {
        Scanner sc= new Scanner(System.in);
        int n= sc.nextInt();
        int q= sc.nextInt();
        int[] arrS = new int[n+1];
        for(int i=1;i<=n;i++){
            arrS[i]=arrS[i-1]+sc.nextInt();
            //System.out.println(arrN[i]);
        }

        for(int i=0;i<q;i++){
            int a= sc.nextInt();
            int b= sc.nextInt();
            System.out.println(arrS[b]-arrS[a-1]);
        }
    }
}
```