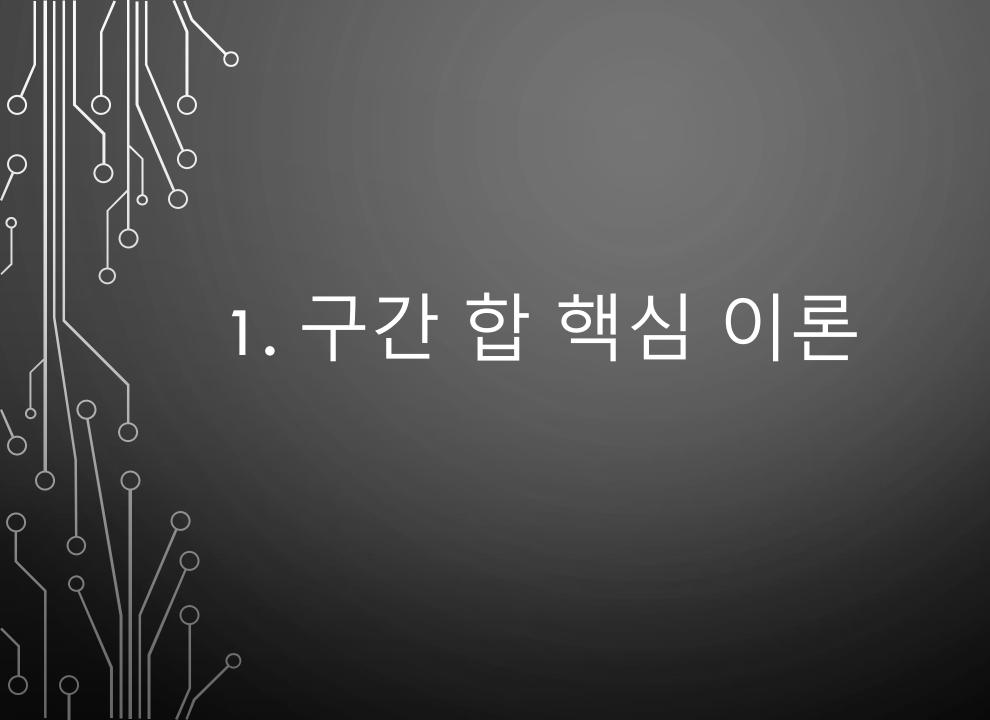
자료구조와 알고리즘

3강 – 누적 구간 합 알고리즘

LECTURED BY SOONGU HONG





- 합 배열은 기존의 배열을 전처리한 배열입니다.
- 합 배열을 미리 구해 놓으면 기존 배열의 일정 범위의 합을 구하는 시간 복잡도가
 O(N)에서 O(1)로 감소합니다.
- A[x] 부터 A[y]까지의 합을 합 배열 없이 구하는 경우 최악의 경우 x가 0이고 y가 N인 경우 시간 복잡도는 O(N)이 됩니다.



• 필수 암기

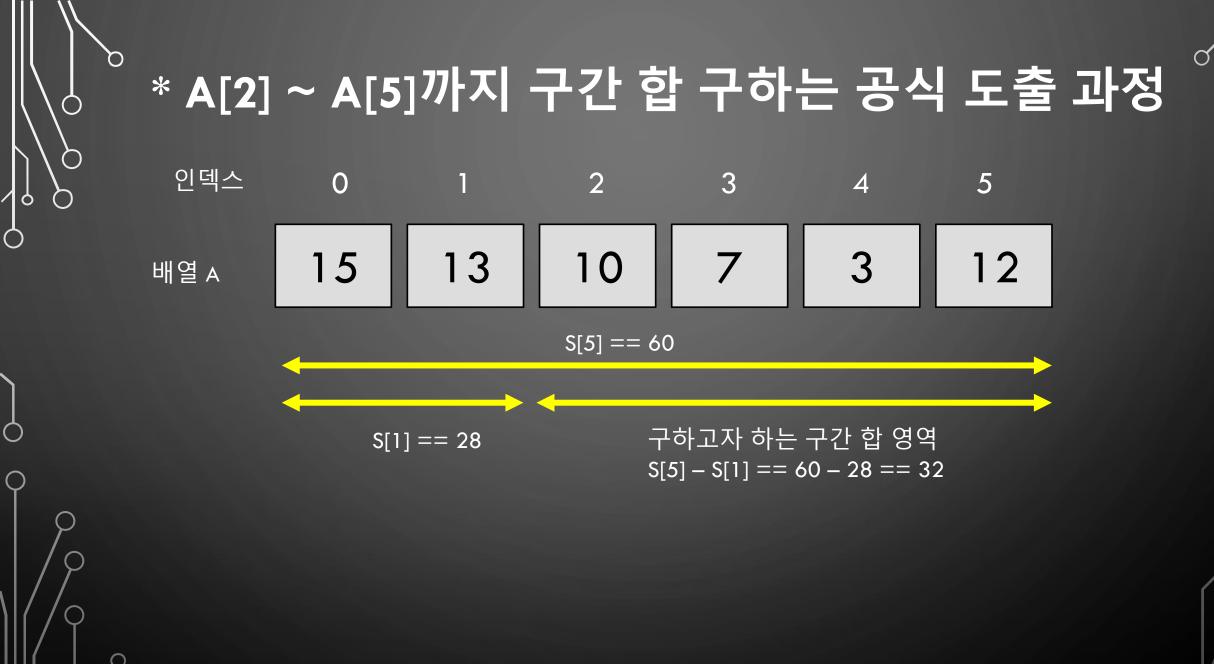
1. 합 배열 S를 만드는 공식

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

2. X ~ Y까지의 구간합을 구하는 공식

$$S[y] - S[x-1]$$

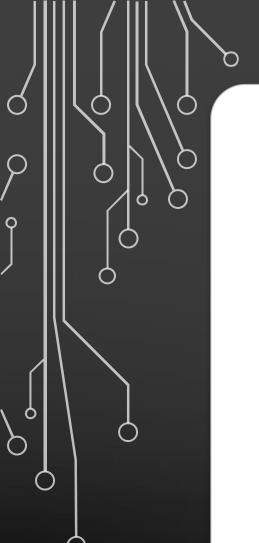
단, x가 0일 경우 -1인덱스가 되는 것은 값을 0으로 처리



2. 핵심 예제 풀어보기

(백준 알고리즘 11659번)

https://www.acmicpc.net/problem/11659



구간 합 구하기 4

☆

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	32798	15219	11934	45.308%

문제

수 N개가 주어졌을 때, i번째 수부터 j번째 수까지 합을 구하는 프로그램을 작성하시오.

입력

첫째 줄에 수의 개수 N과 합을 구해야 하는 횟수 M이 주어진다. 둘째 줄에는 N개의 수가 주어진다. 수는 1,000보다 작거나 같은 자연수이다. 셋째 줄부터 M개의 줄에는 합을 구해야 하는 구간 i와 j가 주어진다.

출력

총 M개의 줄에 입력으로 주어진 i번째 수부터 j번째 수까지 합을 출력한다.

제한

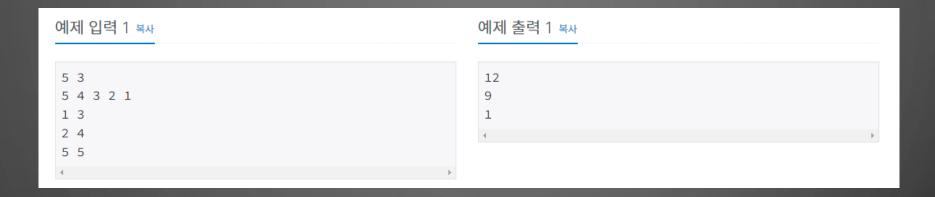
- 1 ≤ N ≤ 100,000
- 1 ≤ M ≤ 100,000
- $1 \le i \le j \le N$

* 문제 분석

- 우선 주어진 수의 개수(N)가 10만개이고, 최대 합을 구해야 하는 횟수(M)가 10만회입니다.
- 구간합을 매번 반복문을 통해 구한다면 최악의 경우 N * M회가 되어서 100억번의 연산이 필요하므로 1초 이내에 수행할 수 없게 됩니다.
- 이럴 때 구간 합 알고리즘을 이용해서 시간 복잡도를 줄여야 합니다.

* 예제 입출력 분석

합배열 S



인덱스 0 1 2 3 4 배열 A 5 4 3 2 1

12



* 예제 입출력 분석

X ~ Y까지의 구간합을 구하는 공식

$$S[y] - S[x-1]$$

* 의사코드 작성하기

```
숫자 개수, 질의 개수 입력 받아 저장하기
for (숫자 개수만큼 반복) {
합 배열 생성하기 (S[i] = S[i-1] + A[i])
}
for (질의 개수만큼 반복) {
질의 범위 입력 받기 (x ~ y)
구간 합 출력하기 (S[y] – S[x-1])
}
```

```
public static void main(String[] args) {
     Scanner sc = new Scanner(System.in);
     int N = sc.nextInt(), M = sc.nextInt();
     long[] S = new long[N + 1];
     for (int \underline{i} = 1; \underline{i} < N + 1; \underline{i} + +) {
          S[\underline{i}] = S[\underline{i} - 1] + sc.nextInt();
     }
     for (int \underline{i} = 0; \underline{i} < M; \underline{i} + +) {
          int x = sc.nextInt(), y = sc.nextInt();
          System.out.println(S[y] - S[x-1]);
```

3. 실전 예제 풀어보기

(백준 알고리즘 11660번)

https://www.acmicpc.net/problem/116560

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	20643	10189	8033	48.688%

문제

N×N개의 수가 N×N 크기의 표에 채워져 있다. (x1, y1)부터 (x2, y2)까지 합을 구하는 프로그램을 작성하시오. (x, y)는 x행 y열을 의미한다.

예를 들어, N = 4이고, 표가 아래와 같이 채워져 있는 경우를 살펴보자.

1	2	3	4
2	3	4	5
3	4	5	6
4	5	6	7

여기서 (2, 2)부터 (3, 4)까지 합을 구하면 3+4+5+4+5+6 = 27이고, (4, 4)부터 (4, 4)까지 합을 구하면 7이다.

표에 채워져 있는 수와 합을 구하는 연산이 주어졌을 때, 이를 처리하는 프로그램을 작성하시오.

입력

첫째 줄에 표의 크기 N과 합을 구해야 하는 횟수 M이 주어진다. $(1 \le N \le 1024, 1 \le M \le 1000,000)$ 둘째 줄부터 N개의 줄에는 표에 채워져 있는 수가 1행부터 차례대로 주어진다. 다음 M개의 줄에는 네 개의 정수 x1, y1, x2, y2 가 주어지며, (x1, y1)부터 (x2, y2)의 합을 구해 출력해야 한다. 표에 채워져 있는 수는 1000보다 작거나 같은 자연수이다. $(x1 \le x2, y1 \le y2)$



- 이 문제 또한 질의의 개수 м이 100,000이므로 질의마다 합을 구해선 안 되고,
 구간 합 알고리즘을 이용해야 합니다.
- 구간 합 배열이 1차원이 아니라 2차원으로 확장된 것을 고민하는 것이 핵심입니다.
- 예제에 나온 4×4 배열을 5×5 로 확장해서 인덱스와 순서의 번호를 일치 시켜 놓고 문제를 해결해 봅시다.

* 예제 입출력 분석

숫자배열 A[i][j]

	1	2	3	4
1	1	2	3	4
2	2	3	4	5
3	3	4	5	6
4	4	5	6	7

구간합배열 D[i][j]

	1	2	3	4
1	1	3	6	10
2	3			
3	6			
4	10			

 \longrightarrow D[1][i] = D[1][i-1] + A[1][i]



D[i][1] = D[i-1][1] + A[i][1]

숫자배열 A[i][j]

	1	2	3	4
1	1	2	3	4
2	2	3	4	5
3	3	4	5	6
4	4	5	6	7

구간합배열 D[i][j]

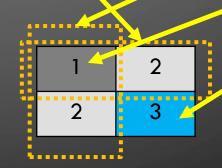
	1	2	3	4
1	1	3	6	10
2	3	8		
3	6			
4	10			

중요!!!

D[i][j]값을 채우는 구간 합 공식

D[i][j] = D[i][j-1] + D[i-1][j] - D[i-1][j-1] + A[i][j]

D[2][2]를 구하려면?? => D[1][2] + D[2][1] - D[1][1] + A[2][2]





완성된 구간합배열 D[i][j]

	1	2	3	4
1	1	3	6	10
2	3	8	15	24
3	6	15	27	42
4	10	24	42	64

질의 2 2 3 4 => (2, 2에서 3, 4)까지의 구간합

	1	2	3	4
1	1	3	6	10
2	3	8	15	24
3	6	15	27	42
4	10	24	42	64

$$D[3][4] - D[1][4] - D[3][1] + D[1][1]$$

=> $42 - 10 - 6 + 1 = 27$

	1	2	3	4
1	1	3	6	10
2	3	8	15	24
3	6	15	27	42
4	10	24	42	64

중요!!!

질의 X1, Y1 부터 X2, Y2까지의 구간합 구하는 공식

D[X2][Y2] - D[X1-1][Y2] - D[X2][Y1-1] + D[X1-1][Y1-1]

* 의사코드 작성하기

```
N(배열 크기) M(질의 수) 입력받아 저장하기
for (N만큼 반복) {
  for (N만큼 반복) {
     원본 배열에 정수 저장하기
for (N만큼 반복) {
  for (N만큼 반복) {
     구간 합 배열 저장하기
     공식 : D[i][j] = D[i][j-1] + D[i-1][j] - D[i-1][j-1] + A[i][j]
for (M만큼 반복) {
  질의 좌표 입력받고 계산 및 출력하기
  공식: D[x2][y2] - D[x1-1][y2] - D[x2][y1-1] + D[x1-1][y1-1]
```