

[Algorithm] 40강 : 구간 합(Interval) 빠르게 구하기 — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2020-12-02 오전 12:18

URL: <https://continuous-development.tistory.com/207>

Algorithm

[Algorithm] 40강 : 구간 합(Interval) 빠르게 구하기

2020. 12. 2. 00:18 수정 삭제 공개

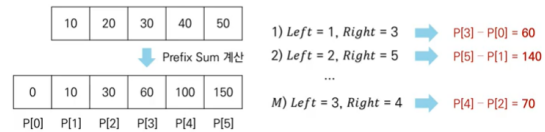
구간 합(Interval)

1.1 구간 합 문제 설명

- 구간 합 문제 : 연속적으로 나열된 N개의 수가 있을 때 특정 구간의 모든 수를 합한 값을 계산하는 문제
- 예를들어 5개의 데이터로 구성된 수열 10,20,30,40,50 이 있다고 가정
 - 두 번째 수부터 네 번째 수까지의 합은 $20 + 30 + 40 = 90$
- N개의 정수로 구성된 수열
- M개의 쿼리 정보가 주어진다
- 각 쿼리는 Left와 Right로 구성
 - 각 쿼리에 대하여 left, right 구간에 포함된 데이터들의 합을 출력
- 수행시간 제한은 $O(N+M)$

1.2 구간 합 빠르게 계산하기

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



1.3 구간 합 구현

```
# 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])
```

'Algorithm' 카테고리의 다른 글

[Algorithm] 40강 : 구간 합(Interval) 빠르게 구하기

[Algorithm] 39강 : 투 포인터(Two Pointers) 알고리즘의 정의와 구현

[Algorithm] 38강 : 에라토스테네스의 체 알고리즘의 정의와 구현

[Algorithm] 37강 : 소수 판별 알고리즘의 정의와 구현□

[Algorithm] 36강 : 위상정렬 알고리즘의 정의와 구현□

[Algorithm] 35강 : 크루스칼 알고리즘의 정의와 구현□

Prefix Sum

구간 합

구간 합 구현



나아무늘보

혼자 끄적끄적하는 블로그 입니다.