



# 16주차: Lazy Propagation in Segment Tree

## 느리게 갱신되는 세그먼트 트리(Lazy Propagation in Segment Tree)

값 변경마저도 segment tree의 성질을 이용해서 더 효율적으로 수행할 수 있다면?

### 필요성 🌟

#### 구간합

- 구간의 값은 고정, 구간의 합 반복해서 구하기

11659번: 구간 합 구하기 4

첫째 줄에 수의 개수  $N$ 과 합을 구해야 하는 횟수  $M$ 이 주어진다. 둘째 줄에는  $N$ 개의 수가 주어진다. 수는 1,000보다 작거나 같은 자연수이다. 셋째 줄부터  $M$ 개의 줄에는 합을 구해야 하는 구간  $i$ 와  $j$ 가 주어진다.

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

BAE<KJOON>  
ONLINE JUDGE

#### 세그먼트 트리

- 구간의 요소가 중간중간 변할 때, 구간의 합 반복해서 구하기

2042번: 구간 합 구하기

첫째 줄에 수의 개수  $N$  ( $1 \leq N \leq 1,000,000$ )과  $M$  ( $1 \leq M \leq 10,000$ ),  $K$  ( $1 \leq K \leq 10,000$ )가 주어진다.  $M$ 은 수의 변경이 일어나는 횟수이고,  $K$ 는 구간의 합을 구하는 횟수이다. 그리고 둘째 줄부터  $N+1$ 번째

<https://www.acmicpc.net/problem/2042>

BAE<KJOON>  
ONLINE JUDGE

- 시간복잡도(길이:  $n$ , 구간 합 연산 횟수:  $m$ , 값 변경 횟수:  $k$ )
  - 트리 생성:  $O(n)$

- 변경:  $O(\log(n)) * m$
- 구간 합 연산:  $O(\log(n)) * k$
- 총:  $O(n + (m + k) * \log(n))$

## 문제 상황

- 구간 내 구간이 중간중간 변할 때, 구간의 합 반복해서 구하기

10999번: 구간 합 구하기 2

어떤 N개의 수가 주어져 있다. 그런데 중간에 수의 변경이 빈번히 일어나고 그 중간에 어떤 부분의 합을 구하려 한다. 만약에 1,2,3,4,5 라는 수가 있고, 3번째부터 4번째 수에 6을 더하면 1, 2, 9, 10, 5가 되

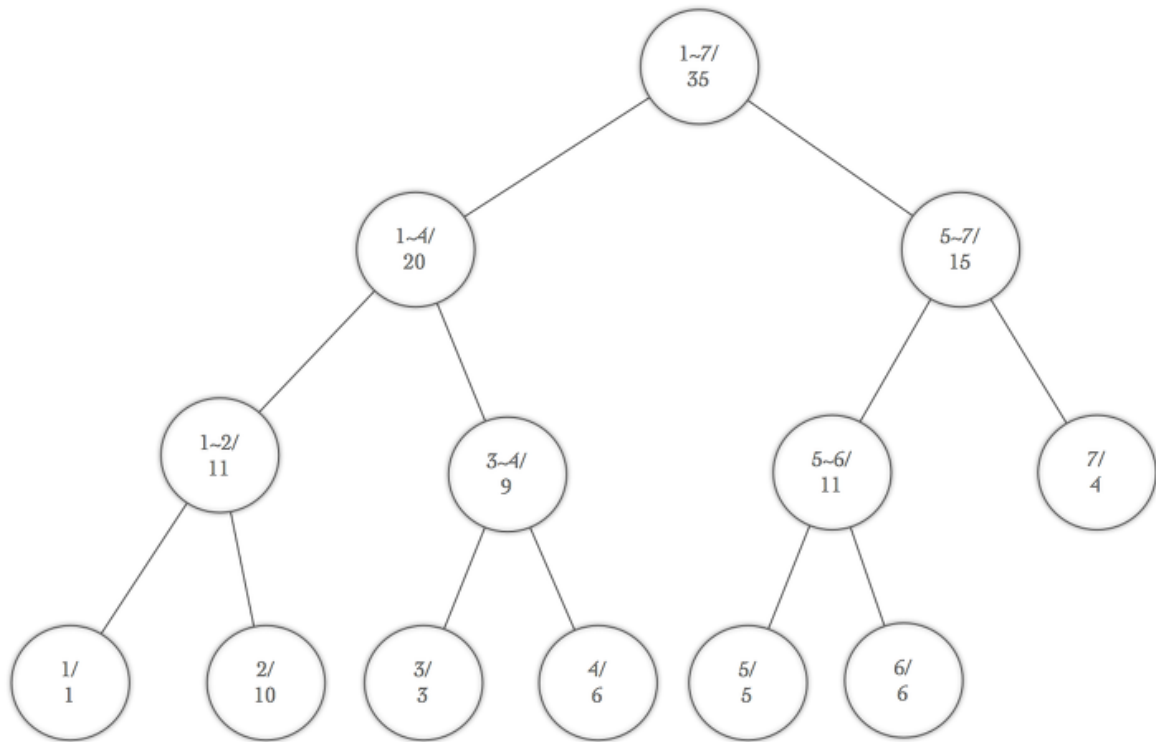
<https://www.acmicpc.net/problem/10999>

BAEKJOON  
ONLINE JUDGE

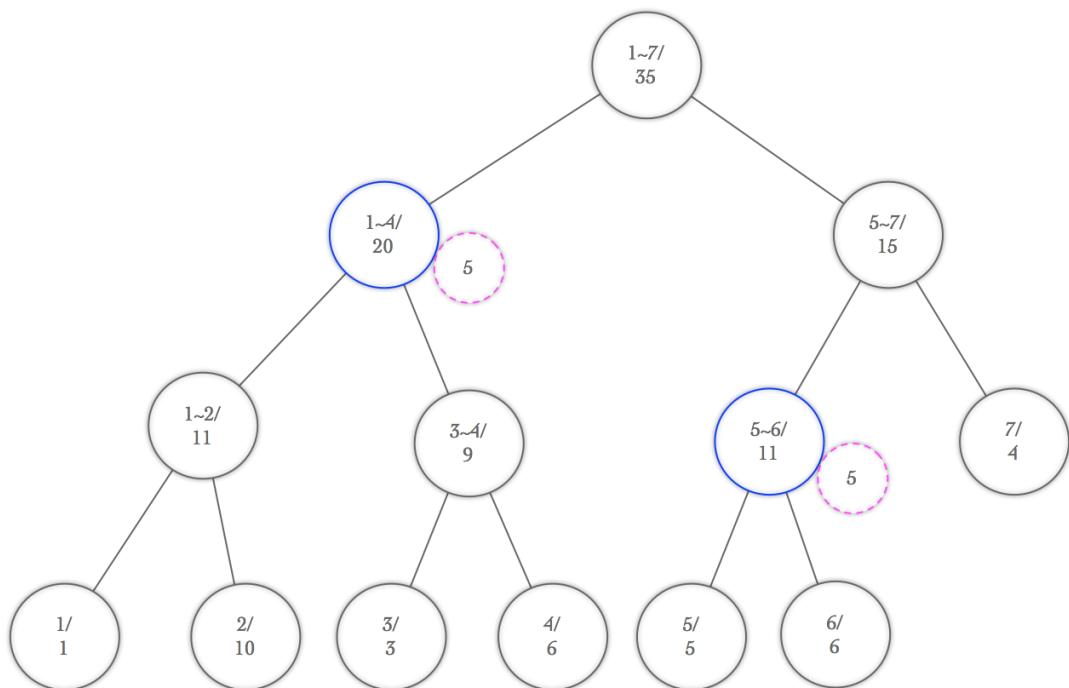
- 시간복잡도(길이: n, 구간 합 연산 횟수: m, 값 변경 횟수: k, 값 변경 구간 크기: p)
  - 트리 생성:  $O(n)$
  - 구간 합 연산:  $O(\log(n)) * m$
  - 변경:  $O(\log(n)) * k * p$
  - 총:  $O(n + (m + kp) * \log(n)) \rightarrow kp$  항으로 인해 TLE

## Segment Tree?

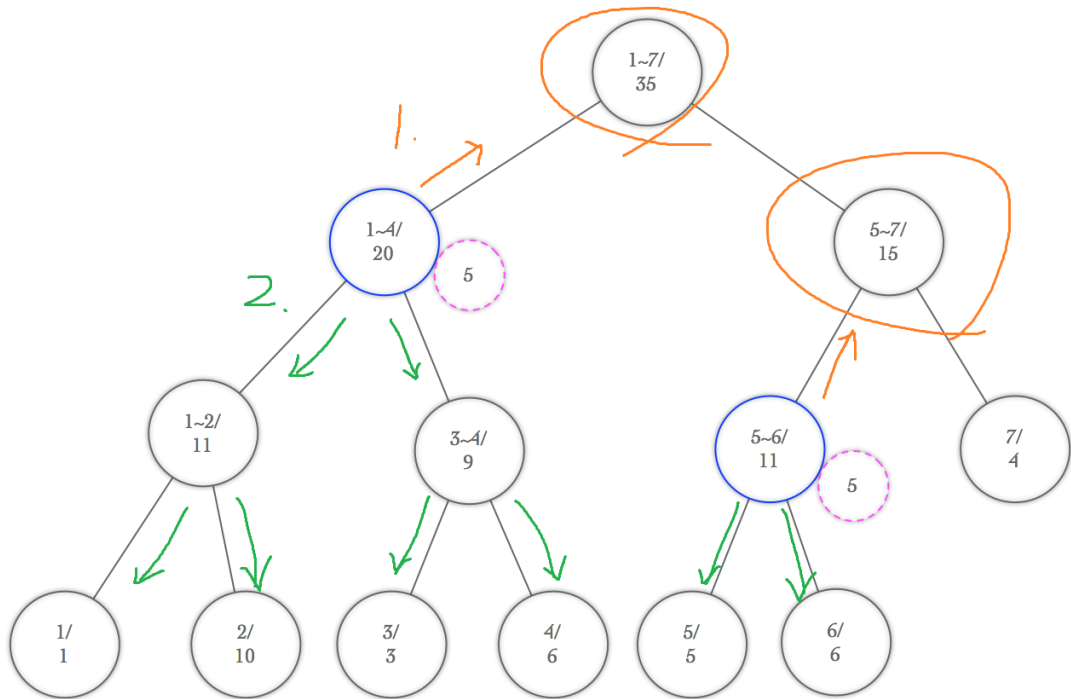
Segment tree는 구간의 정보를 기록하는 tree이다.



- 아래서 올라가는 방식이 아닌, 세그먼트 트리의 성질을 이용해서 구간의 정보를 변경하자!
- ex) 1~6 구간에 +5를 하는 경우



- 이 다음은?



- 아래를 갱신할 필요가 있을까?
  - 값 변경이나 합 구하기나 모두 root에서 내려가는 방식
  - 기록만 해 두고, 아래까지 내려가는 경우에 갱신하자!
  - 즉, 느리게 전파하자!

## 시뮬레이션

### 느리게 갱신되는 세그먼트 트리

세그먼트 트리 에 설명한 문제에서 2번 연산은 \$\$\$번째 수에 \$\$\$를 더하는 것이었습니다. 이번 문제의 2번 연산은 구간에 수를 더하는 것입니다. 세그먼트 트리에서 수를 변경하는 연산을 \$\$\$번째 수부터 \$\$\$번

<https://book.acmicpc.net/ds/segment-tree-lazy-propagation>

BOJBOOK

## 오늘의 문제

- 없음
  - Lazy propagation와 segment tree의 개념적 이해가 우선
  - 따라서 구현에 관련된 부분은 단계별로 자신에게 필요한 단계에 대한 학습을 하는 것을 추천

- 단계별 학습 목록
  - 0단계: Tree 형태의 자료구조를 만들고, 사용하는 인터페이스를 만들어 보자!
    - 클래스와 메소드 형태로 만들면 더욱 좋을 것이다!
    - 참조: <https://www.delftstack.com/ko/howto/python/trees-in-python/>
  - 1단계: 일반적인 list를 완전 이진 트리로 사용해 보자!
    - 앞서 다뤘던 tree 형태의 자료구조들에 대한 내용 참고
    - 참조: <https://doing7.tistory.com/63>
  - 2단계: Segment tree 구현
    - 13주차에 다뤘던 segment tree를 구현해 보자!
    - <https://www.notion.so/13-Segment-Tree-286fd85daed548f3911d7e93f6a57a78>
  - 3단계: Lazy propagation 구현
    - 문제를 풀면서 느린 전파를 구현해 보자!
    - 구간 합 구하기2(Platinum 4): <https://www.acmicpc.net/problem/10999>
    - 참조: <https://book.acmicpc.net/ds/segment-tree-lazy-propagation>

## 참고 자료

- 블로그: <https://bowbowbow.tistory.com/4>
- 백준: <https://book.acmicpc.net/ds/segment-tree-lazy-propagation>