

Fenwik Tree

V2017117 강현민

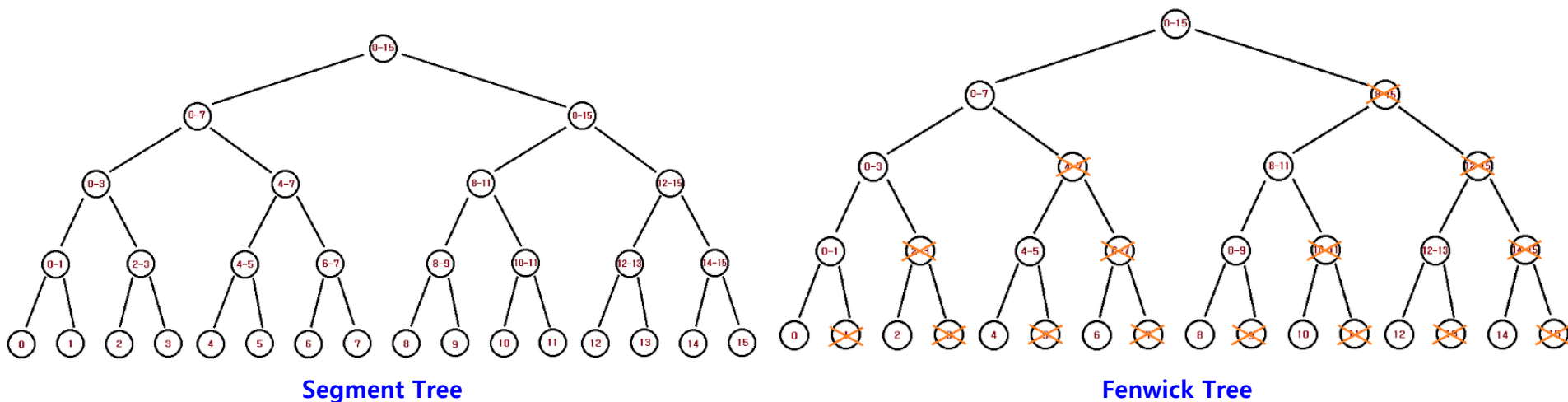


서강대학교 영상대학원
SOGANG UNIVERSITY GRADUATE SCHOOL OF MEDIA

Fenwick Tree(Binary Index Tree)

Introduction

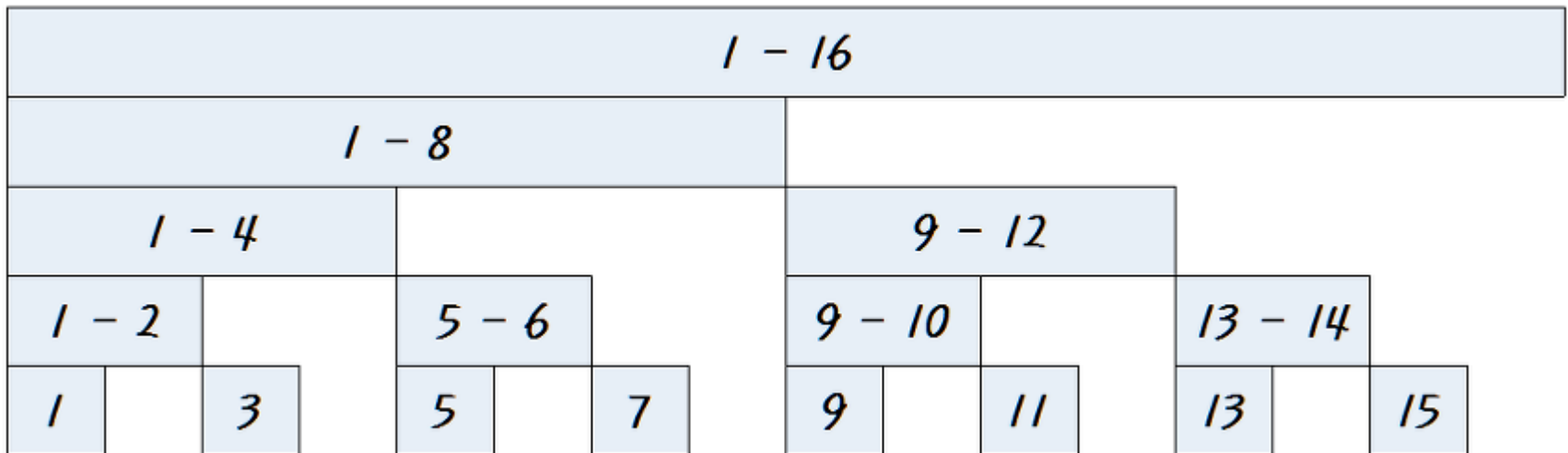
- 펜윅 트리는 세그먼트 트리의 메모리를 더 절약하기 위해 만든 방법.
- 펜윅 트리는 세그먼트 트리의 변형으로 복잡도는 $\log N$ 만큼 수행 한다.
- 펜윅 트리의 핵심은 구간 합을 부분 합만으로 빠르게 계산할 수 있는 자료 구조를 만들어 우리가 원하는 구간 합을 계산 할 수 있다는 것이다.



Fenwick Tree(Binary Index Tree)

Fenwick Tree

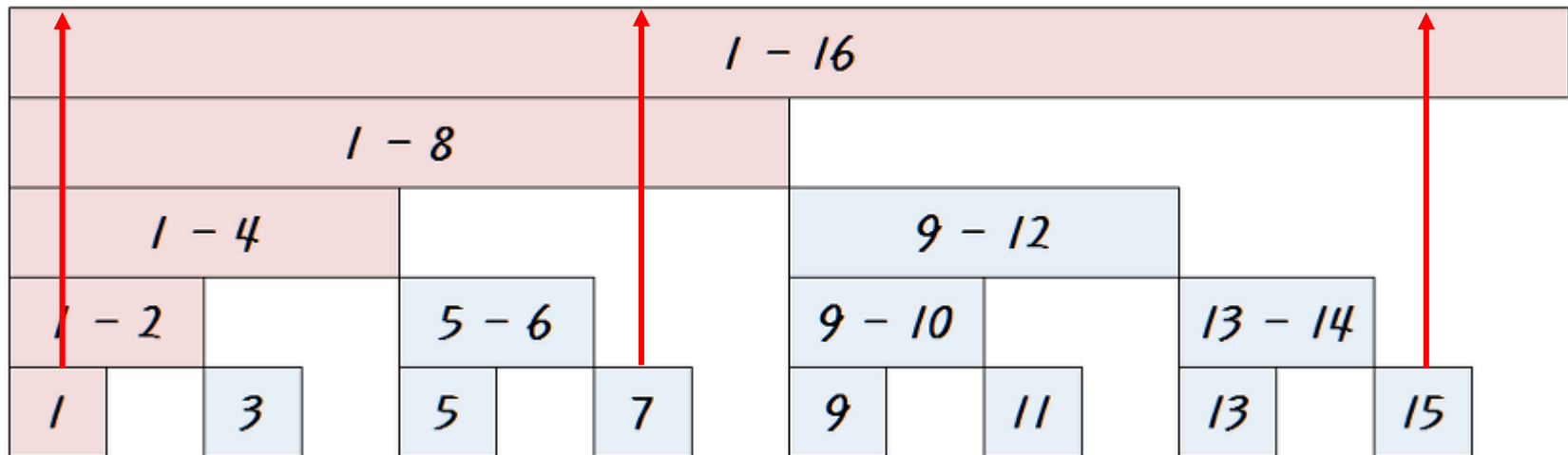
- 이전의 그림을 아래와 같이 변형 할 수 있다.
- 펜윅 트리는 비트를 이용하기에 인덱스를 0이 아닌 1번 부터로 바꾼다.(+1)



Fenwick Tree(Binary Index Tree)

펜윅 트리 특성

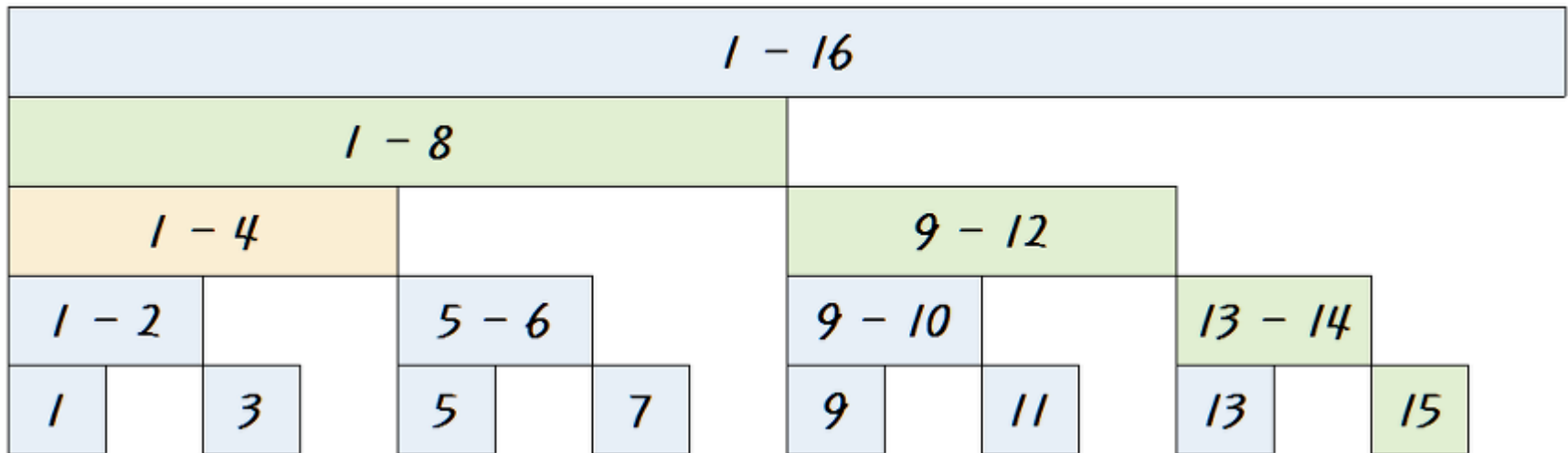
- 예를 들어 1번 인덱스의 값을 변경 했다면 1, 1-2, 1-4, 1-8, 1-16을 업데이트 해준다.
- 만약 7번을 변경 한다면 7, 1-8, 1-16을 업데이트 해주면 된다.
- 15번을 변경 하면 15, 1-16을 업데이트 해주면 된다.



Fenwick Tree(Binary Index Tree)

펜윅 트리 특성

- 만약 5~15번 구간의 합을 구하고 싶다면 어떻게 해야 할까?
- 아래 그림과 같이 $(1-8) + (9-12) + (13-14) + 15$ 에 겹치는 구간 $(1-4)$ 를 빼주면 구간 합을 얻을 수 있게 된다.
- $1\sim 7$ 까지의 구간 합은 $(1-4) + (5-6) + 7$ 을 이용하여 구간 합을 구하게 된다.



Fenwick Tree(Binary Index Tree)

펜윅 트리의 비트 처리 방식

- 1~13의 구간의 합은 아래 그림과 같다.
- $1101 + 1100 + 1000$ 와 같은데 특징을 살펴보면 가장 오른쪽 비트가 0으로 바뀐다.
- 예를 들어 $val=1100$ 이 있다면 $\sim val=0011$ 이며 여기에 $+1$ 을하면 0100 이다.
- 결국 0100 은 $-val$ 을 의미하고 여기서 1은 최하위 비트의 위치를 의미한다.
- 이것을 식으로 만들면 $val -= (val \& -val)$ 로 나타낼 수 있다.
- 업데이트는 반대로 최하위 비트에 $+1$ 을 하면 된다.
- 즉, $val += (val \& -val)$ 로 정의할 수 있다.

