Fenwick Tree

Fenwick Tree

- i의 마지막 자리 : i를 2진수로 나타내었을 때, 가장 마지막 1이 나타내는 값

3 = 11_2 → 1

10 = 1010_2 → 2

12 = 1100_2 → 4

- ~N + 1 == -N

  ex) N = 01001100

  ~N = 10110011

  ~N+1 = 10110100 : N의 보수 + 1 → N의 음수

- N & -N : 마지막 비트 참조

  ex) N = 01001100

  & -N = 10110100

  N &(-N) = 00000100

Fenwick Tree

Fenwick Tree

- tree[i] : i의 가장 마지막 비트 만큼 i부터 그 앞까지 합을 저장한다.

  13 = 1101_2 : tree[13] = A[13] ~ A[13]까지의 합

  12 = 1100_2 : tree[12] = A[9] ~ A[12]까지의 합

  8 = 1000_2 : tree[8] = A[1] ~ A[8]까지의 합

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	2	1	4	1	2	1	8	1	2	1	4	1	2	1	16

1		3		5		7		9		11		13		15	
2				6				10				14			
4								12							
8															
16															

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]	A[11]	A[12]	A[13]	A[14]	A[15]	A[16]
3	2	5	7	10	3	2	7	8	2	1	9	5	10	7	4

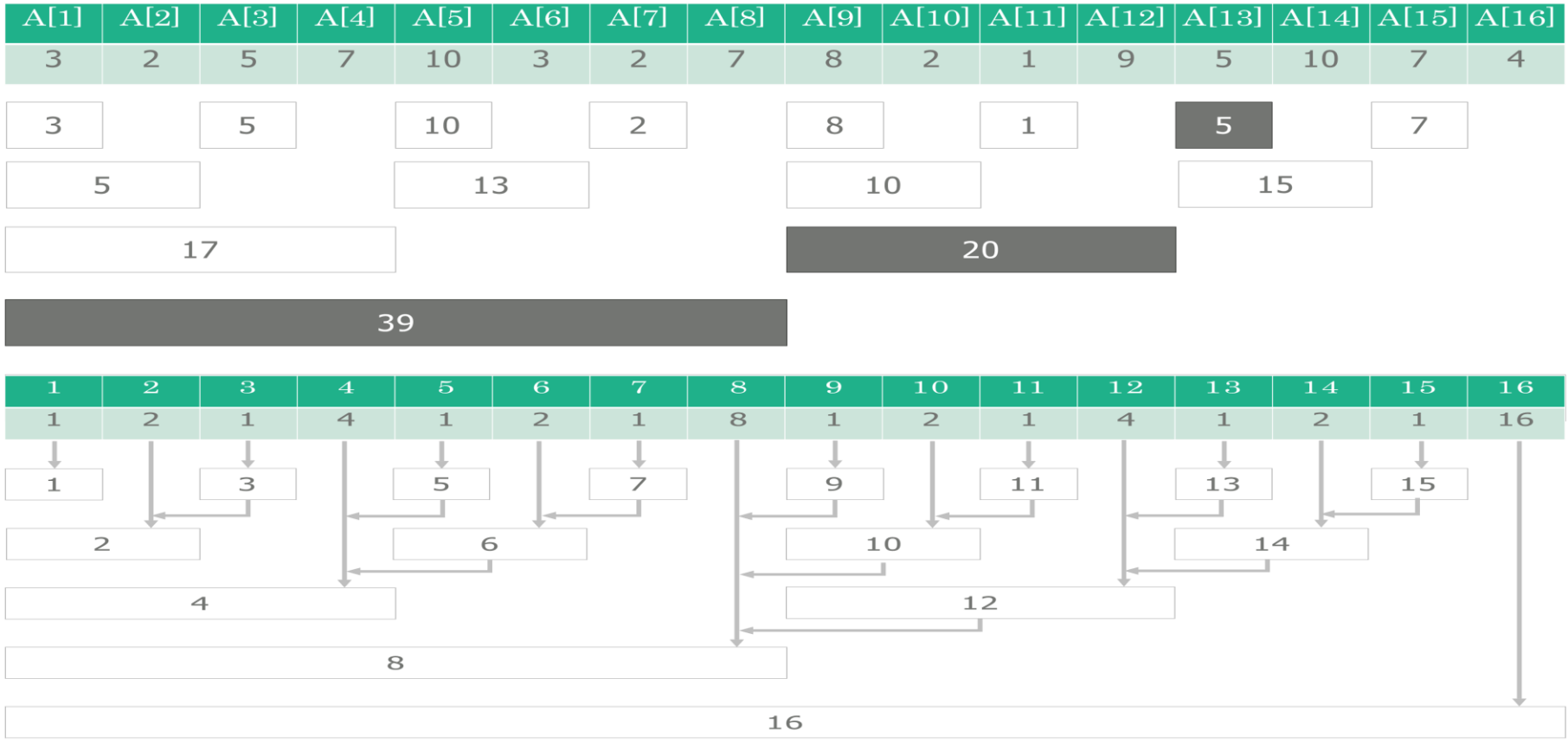
3		5		10		2		8		1		5		7	
5				13				10				15			
17								20							
39															
85															

Fenwick Tree

Fenwick Tree

- A[1] + … + A[13]을 구하려면 13 = 1101_2

  => tree[1101_2] + tree[1100_2] + tree[1000_2] : 가장 마지막 1이 0이 되어 0_2까지 반복



Fenwick Tree

Fenwick Tree

```
int range_sum(int x) {
        int ans = 0;
        for (int i = x; i > 0; i -= i & -i) {
                ans += tree[i];
        }
        return ans;
}
```

Fenwick Tree

Fenwick Tree

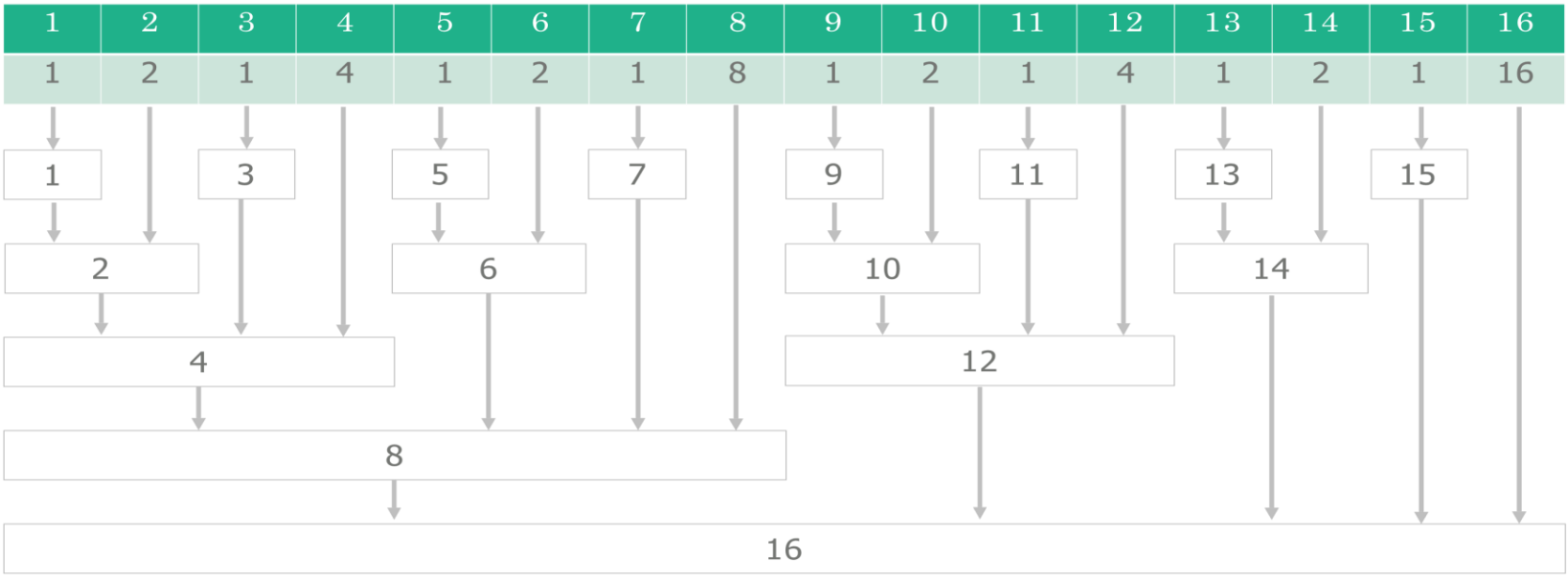
- update를 하려면 변경 분 만큼 변경되는 원소와 변경되는 원소의 마지막 비트를 변경되는 원소에 더한 값에 계속 더해주면 된다.

  ex) 3번째 원소가 변경이 되면 변경되어야 하는 대상은 3 → 4 → 8 → 16이 된다.

  3 = 11_2의 마지막 비트 1_2를 3에 더해주면 4 = 100_2 이 된다.

  4 = 100_2의 마지막 비트 100_2을 4에 더해주면 8 = 1000_2이 된다.

  8 = 1000_2의 마지막 비트 1000_2을 8에 더해주면 16 = 10000_2이 되고 최대 인덱스와 같으므로 종료한다.



Fenwick Tree

Fenwick Tree

```
int update(int x, int diff) {
        for (int i = x; i <= N; i += i & -i) {
                tree[i] += diff;
        }
}
```