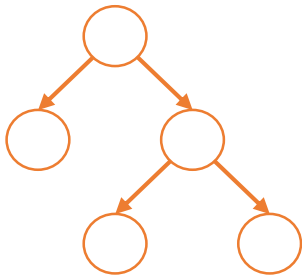


이진 트리의 순회

■ 재귀를 사용한 순회



```
find(n)
{
    Visit(n);
    if Left
        find(Left);
    if Right
        find(Right);
}
```

```
find(n)
{
    if Left
        find(Left);
    Visit(n);
    if Right
        find(Right);
}
```

```
find(n)
{
    if Left
        find(Left);
    if Right
        find(Right);
    Visit(n);
}
```

```
find(n)
{
    if(n)
        Visit(n);
        DLR(Left);
        DLR(Right);
}
```

```
find(n)
{
    if(n)
        DLR(Left);
        Visit(n);
        DLR(Right);
}
```

```
find(n)
{
    if(n)
        DLR(Left);
        DLR(Right);
        Visit(n);
}
```

세그먼트 트리

- n 개의 정수에 대한 일정 구간의 합을 포화이진 트리에 저장.
 - 각 노드에는 구간의 시작과 끝 인덱스와 구간의 합을 저장한다.
 - 루트 노드에 전 구간 인덱스와 합이 저장된다.
 - 자식 노드에는 부모가 가진 구간의 절반씩을 저장.
 - 부모 노드의 구간 경계가 $s = i, e = j$ 인 경우
 - 왼쪽 자식 노드의 구간 $s = i, e = (i+j)/2$.
 - 오른쪽 자식 노드의 구간 $s = (i+j)/2+1, e = j$.
 - 구간의 길이가 1이면 잎 노드.
 - 포화이진트리에서 부모 노드 번호가 n 인 경우.
 - 왼쪽 자식 : $2*n$
 - 오른쪽 자식 : $2*n + 1$

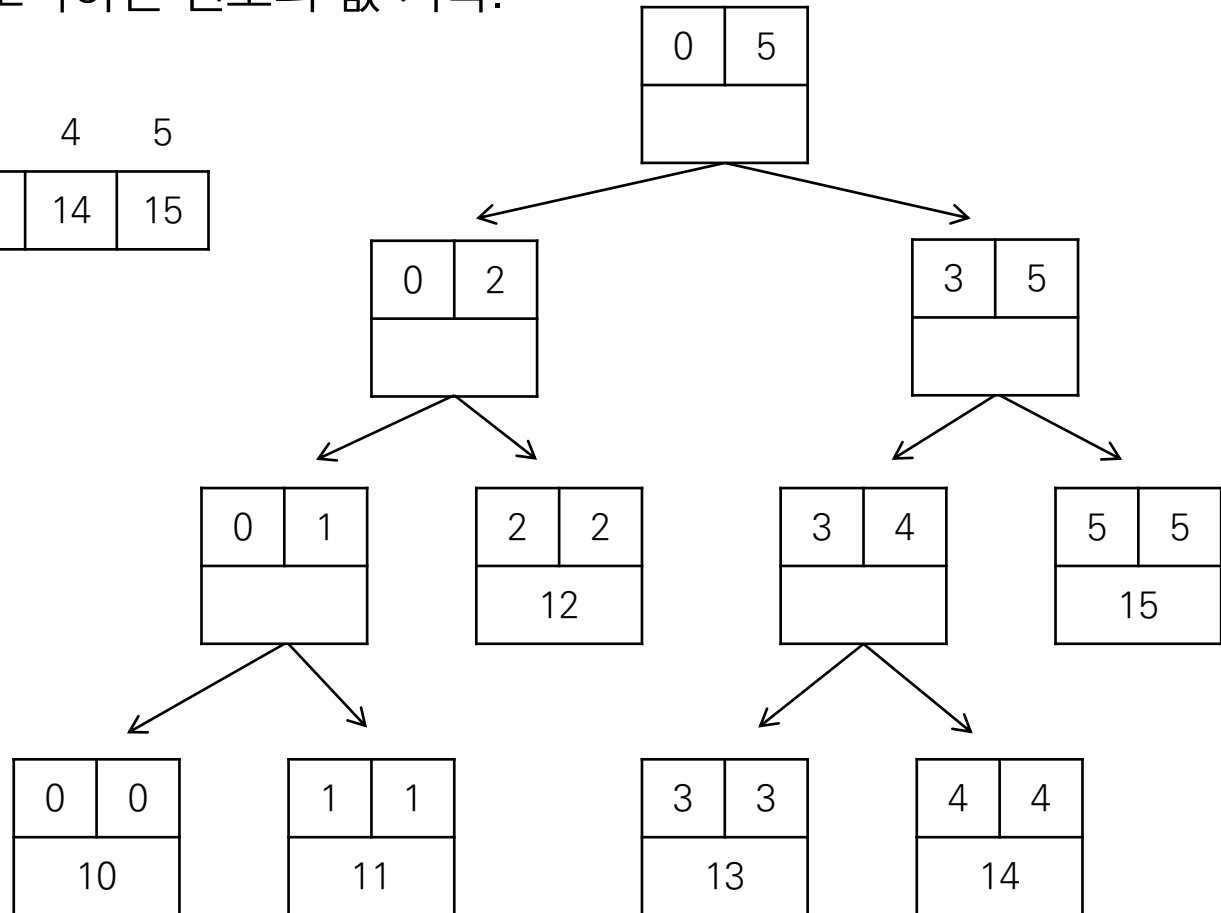
node

| | |
|-----|---|
| s | e |
| sum | |

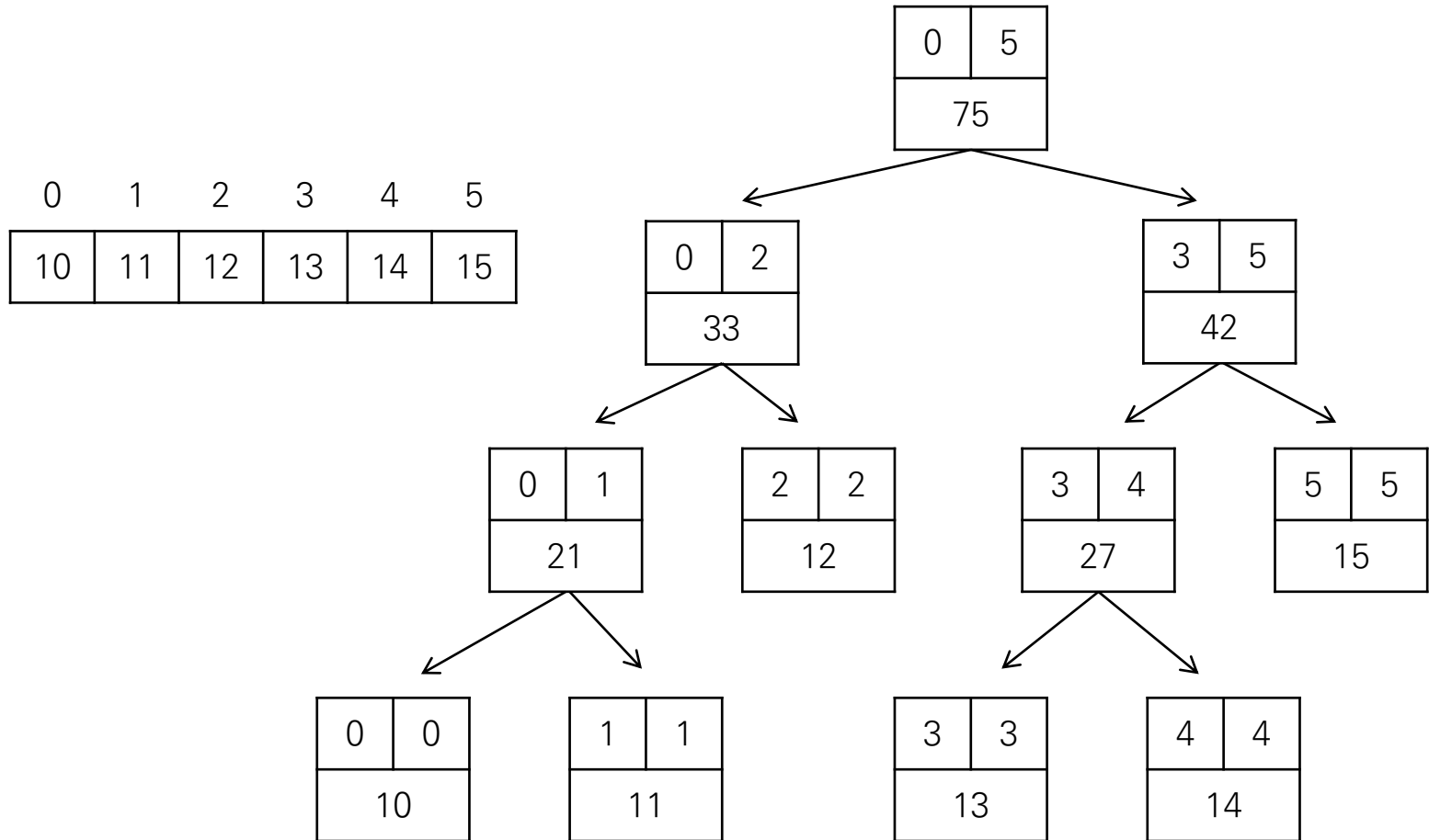
■ 전위 순회로 구간을 나눔.

- 앞 노드에 도착하면 원소의 값 기록.

| 0 | 1 | 2 | 3 | 4 | 5 |
|----|----|----|----|----|----|
| 10 | 11 | 12 | 13 | 14 | 15 |



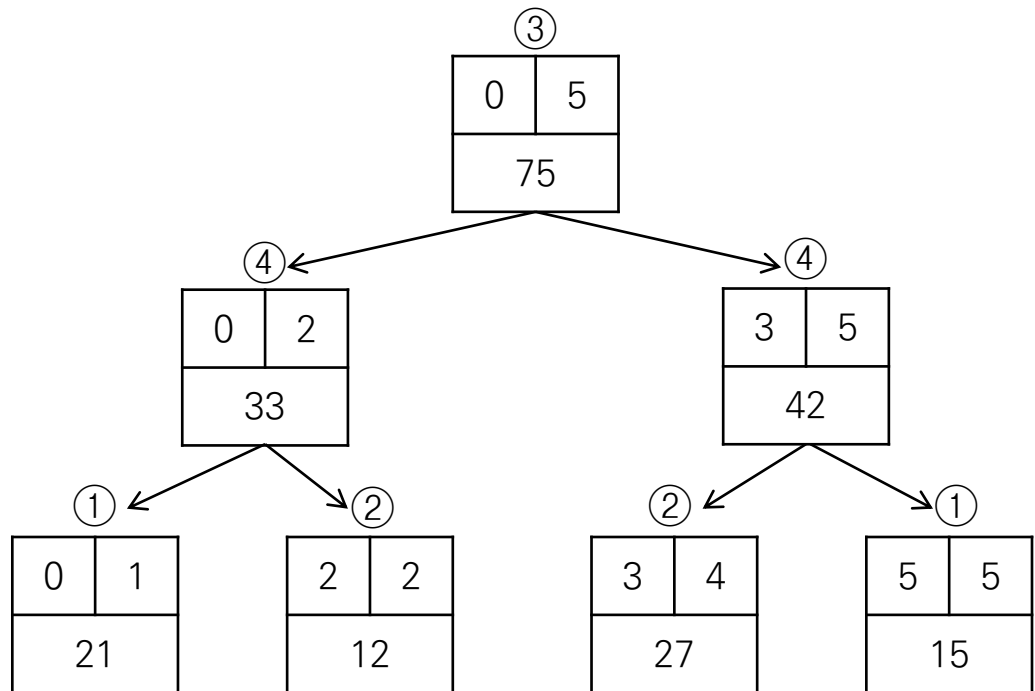
- 후위 순회로 자식 노드의 구간 합을 계산.



■ L-R 구간의 합을 구하기 위한 탐색 과정.

- ① if ($R < s \parallel e < L$) return 0 : 범위 밖.
- ② if ($R \leq s \ \&\& \ e \leq L$) return sum : 유효한 구간
- ③ if ($s \leq L \ \&\& \ R \leq e$) 계속 탐색 후 리턴 값끼리 더함.
- ④ 일부 구간 겹침 : 계속 탐색 후 리턴 값끼리 더함.

| | | L | | R | |
|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 |
| 10 | 11 | 12 | 13 | 14 | 15 |



■ i값을 m으로 변경.

① if ($i < s \parallel e < i$) return 0 : 범위 밖.

② if ($s \leq i \ \&\& \ i \leq e$) 계속 탐색 후 리턴 값의 합을 sum에 더함.

③ if ($s == i \ \&\& \ i == e$)

$diff = m - sum$

$sum = m$

return diff

| i | | | | | |
|----|----|---------|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 |
| 10 | 11 | 12 5 | 13 | 14 | 15 |

