Basics of Segment Tree

Friday, 14 January 2022 9:06 PM

A → [10, 2, 6, -3, 5, 8, 1, 15] (indices 0–7)

1 ×

$N = 10^6$

Find min element from index 2 to 5 → -3

Find max element from index 1 to 6 → 8

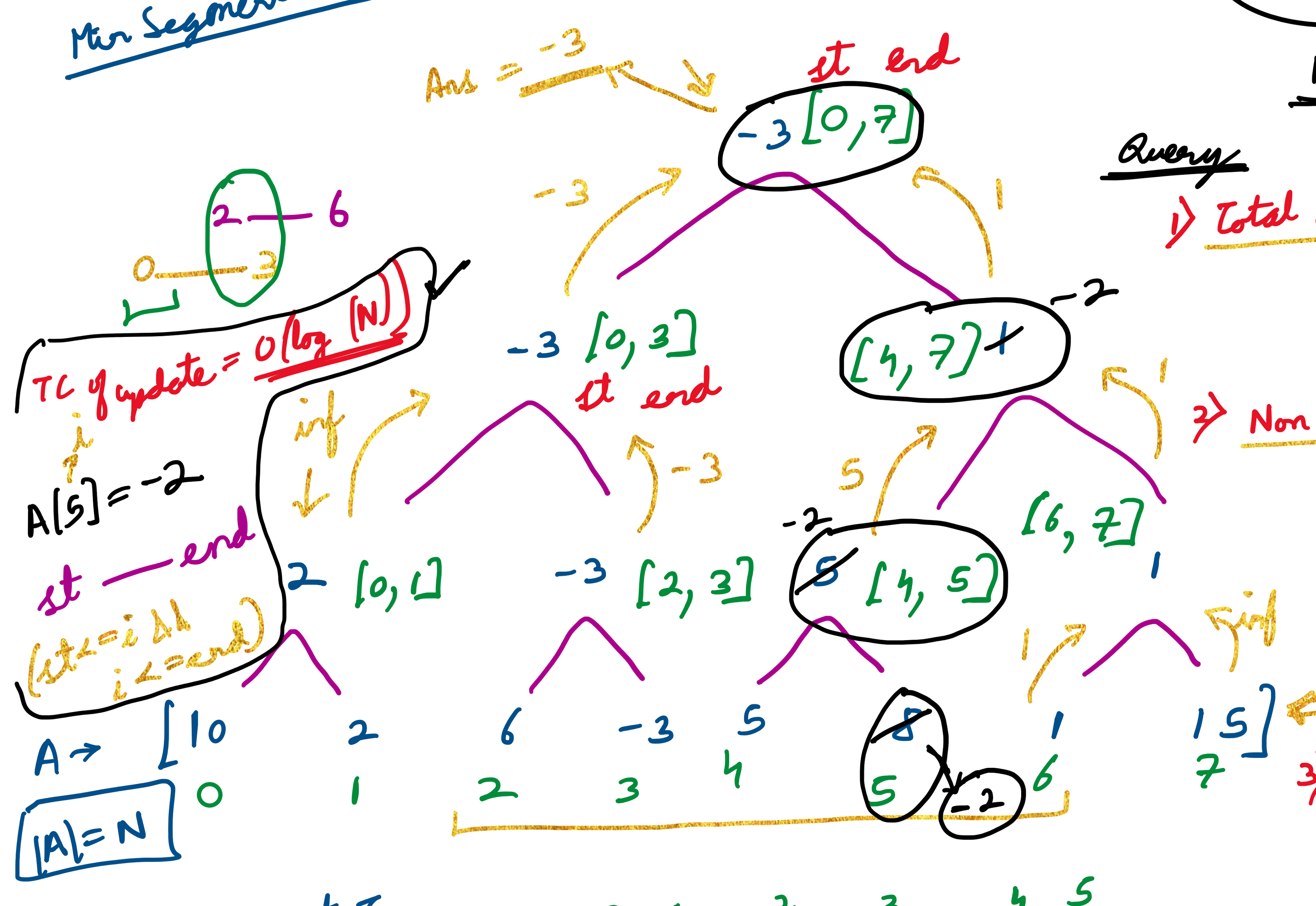
Find sum of elements from index 3 to 5 → 10

1 —— 20 —— 10^6
log N   N

Bruteforce → linear traversal over the range ⇒ TC = O(Range) ≃ O(N)

Multiple Queries (Q) → Find – for a range ⇒ TC = O(N * Q) ← (Q * log(N))
→ Update A[i] = x ⇒ TC = O(1) ← (Q * log(N))

Min Segment Tree



Min (2, 6) → L, R

Query

1) Total overlap → use data ✓

L st end R

2) Non-overlap → ignore data ✓

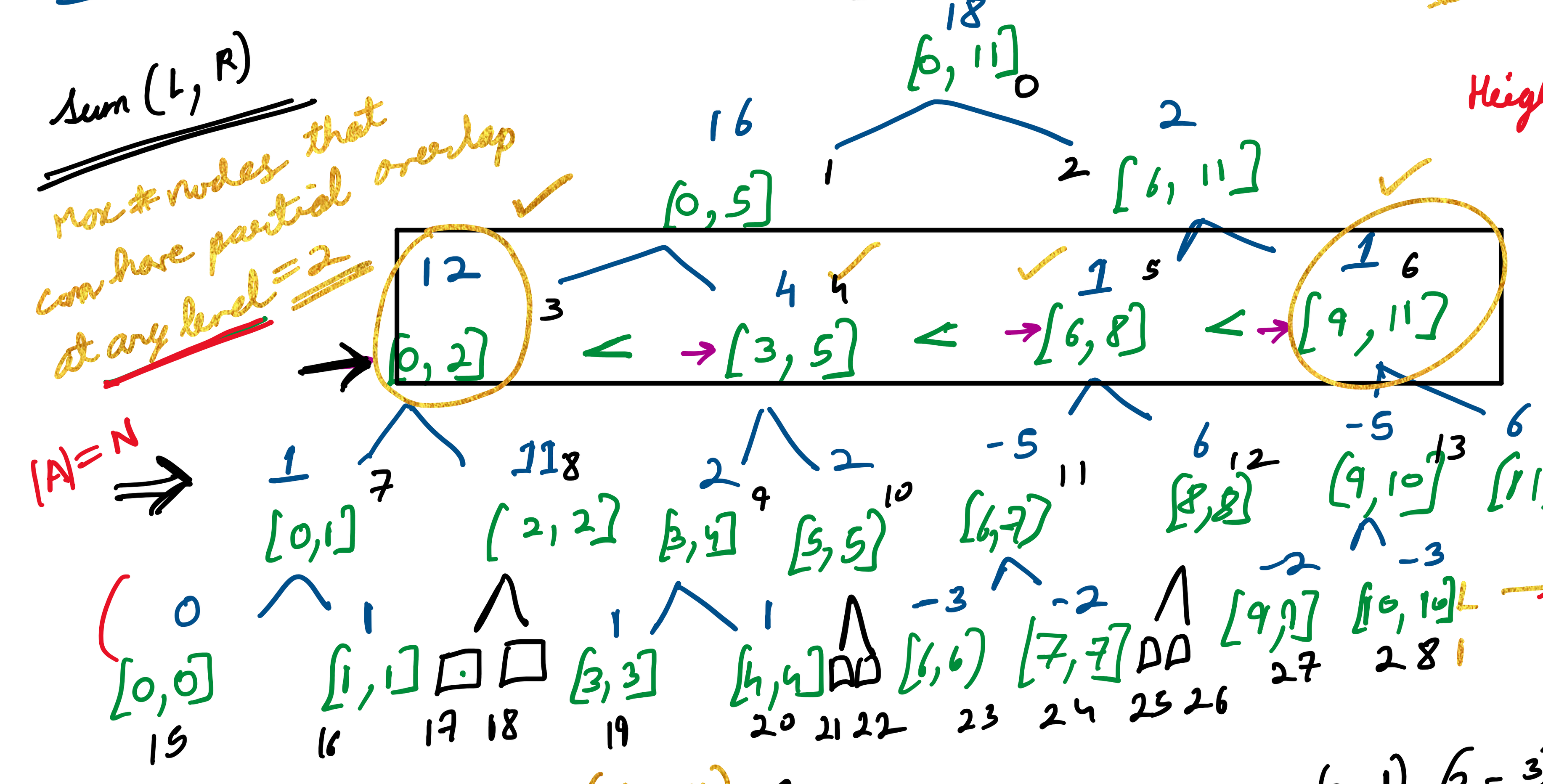
L R st end

st end L R

3) Partial overlap ✓

go to children

Sum Segment Tree



Height = O(log(N))

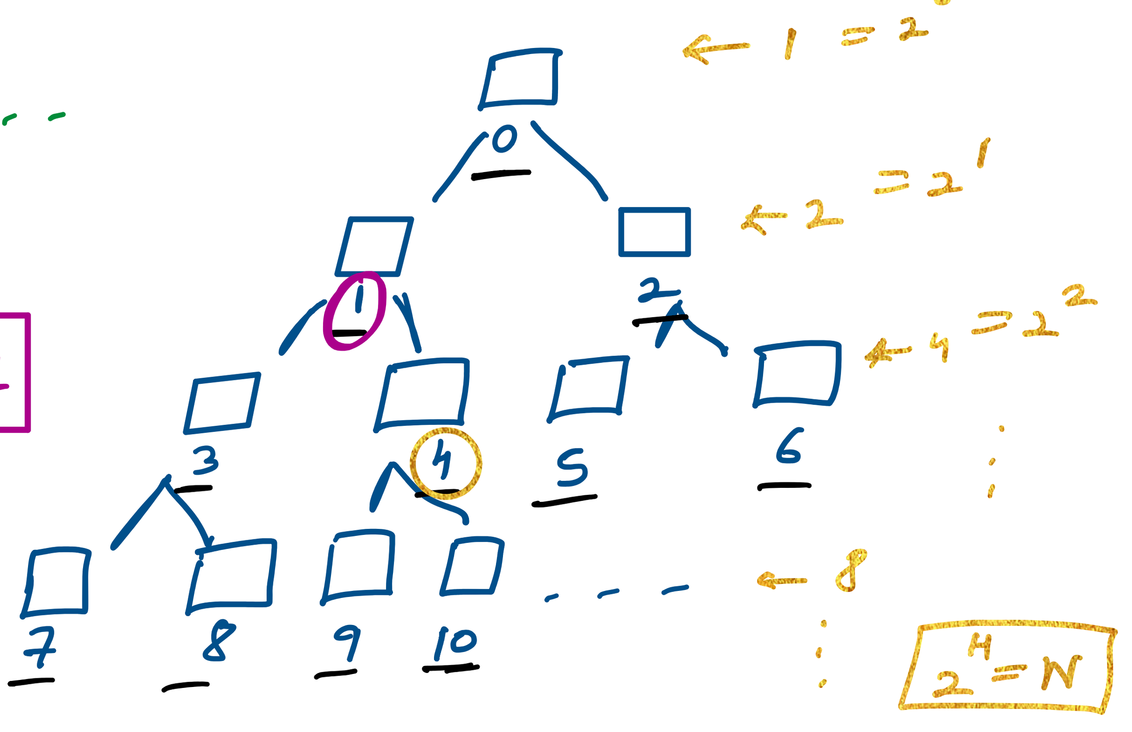
10 → 2N

Unused space < 2N

(0–1) (2–3) (4–5) . . .

TC per query = O(log N) ✓

Build Segment Tree



i → left child = 2 * i + 1

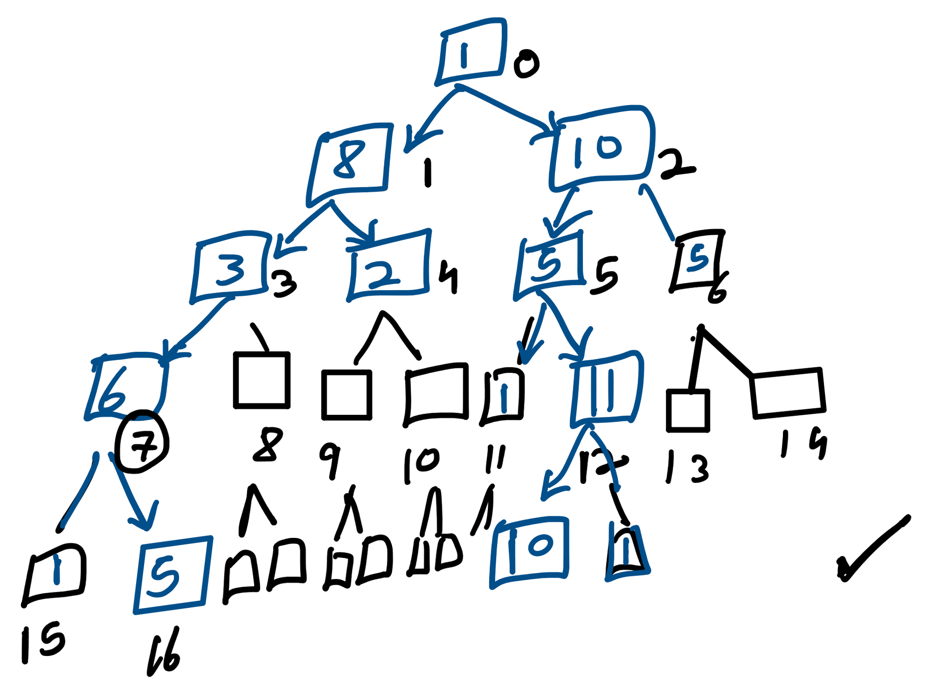
→ Right child = 2 * i + 2

→ Parent = (i-1)/2

rc = lc + 1

i = 1   lc = 2*1+1 = 3
parent = (1-1)/2 = 0   rc = 2*1+2 = 4

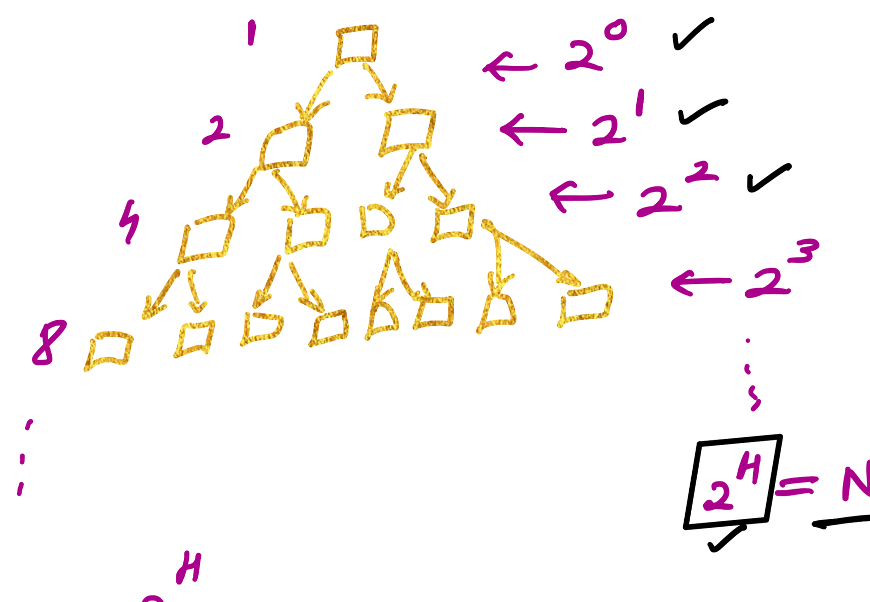
i = 4   lc = 2*4+1 = 9
rc = 2*4+2 = 10
parent = (4-1)/2 = 3/2 = 1



|A| = N

$\boxed{|tree[\,]| = 4N}$

SC = O(N) ✓



$\boxed{2^H} = N$ ✓

$\Rightarrow 1 + 2 + 4 + 8 + \dots 2^H$

$\Rightarrow 2^0 + 2^1 + 2^2 + \dots 2^H = \frac{2^0(2^{H+1} - 1)}{2-1} = \boxed{2^{H+1} - 1} = 2 \cdot 2^H - 1 = \boxed{2N - 1}$ ✓

Build

```
void build(idx, st, end) {          // idx → index in tree array
    if (st == end) { tree[idx] = A[st]; }    // leaf node ✓
    else {
        mid = (st + end) / 2;
        lc = 2 * idx + 1;   rc = 2 * idx + 2;
        build(lc, st, mid);   build(rc, mid+1, end);
        tree[idx] = min(tree[lc], tree[rc]);   ✓
} }
```

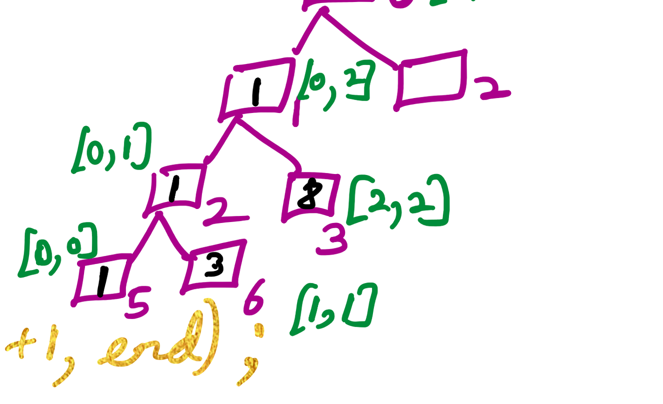
idx → 0, st → 0, end → N-1

min → max / sum / ^ / & / | / gcd

mid =
st = 2
end = 2
idx = 3

lc =
rc =

A → [1, 3, 8, 2, 5]



TC = O(N) ✓

Query

```
int query(idx, st, end, L, R) {
    if (L <= st && end <= R) return tree[idx];   // complete overlap
    if (L > end || st > R) return inf;           // non-overlap
    mid = (st + end) / 2;
    lc = 2 * idx + 1;         rc = 2 * idx + 2;
    return min( query(lc, st, mid, L, R),        // partial overlap
                query(rc, mid+1, end, L, R));
}
```

idx → 0, st → 0, end → N-1, L, R → i/p

L st end R

L R st end

st end L R

TC = O(log(N))