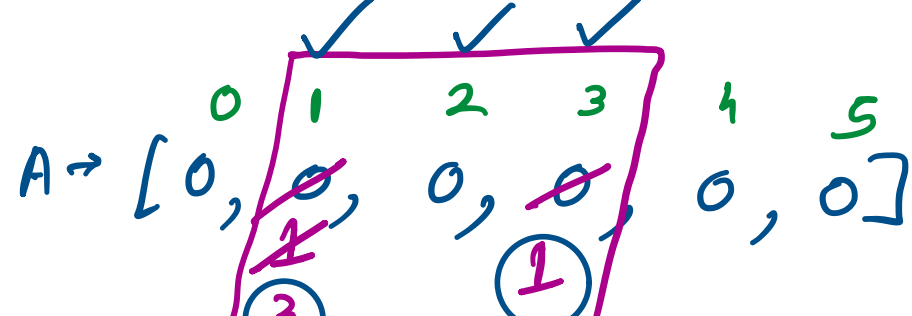


Q1 → Bob & Queries

Given an integer array where all values are 0 initially.

- Queries
- 1 x → $A[x] = A[x] * 2 + 1$
 - 2 x → $A[x] = \lfloor A[x] / 2 \rfloor$
 - 3 x y → Find no. of 1's in binary representation from index x to y.



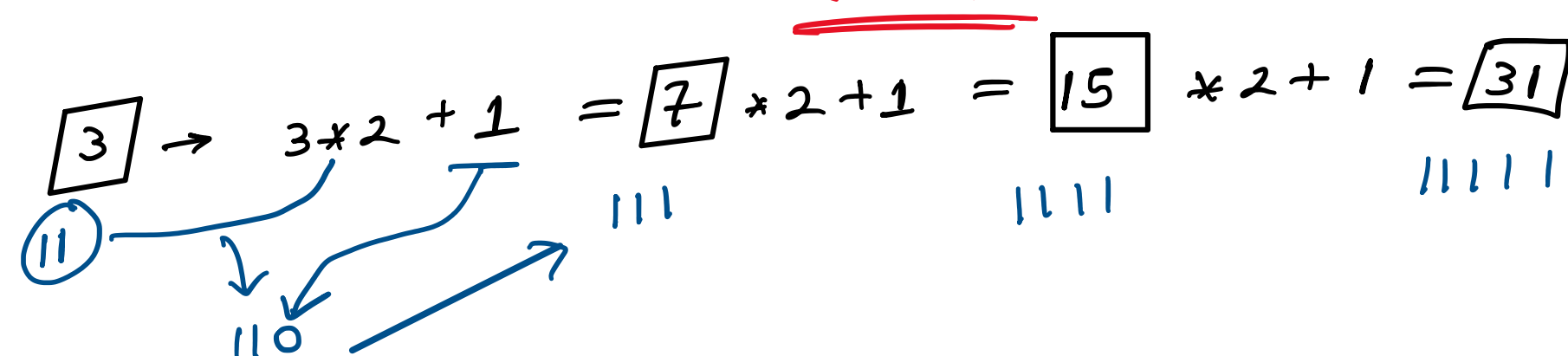
3 → (11)
0 → 0
1 → (1)

Ans = 3

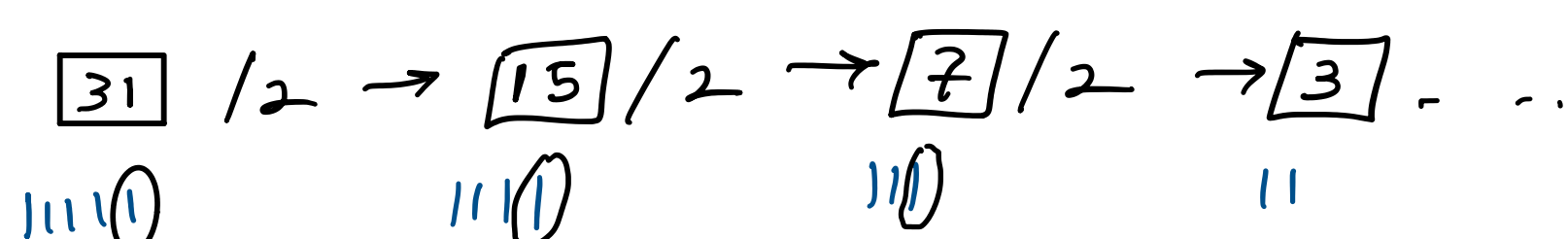
Brute → Q1 & Q2 → simple update $TC = O(1)$

Q3 → Travel from index x to y $TC = O(N)$

$TC = O(Q * N)$



Q1 → Add 1 bit in binary representation of $A[x]$.

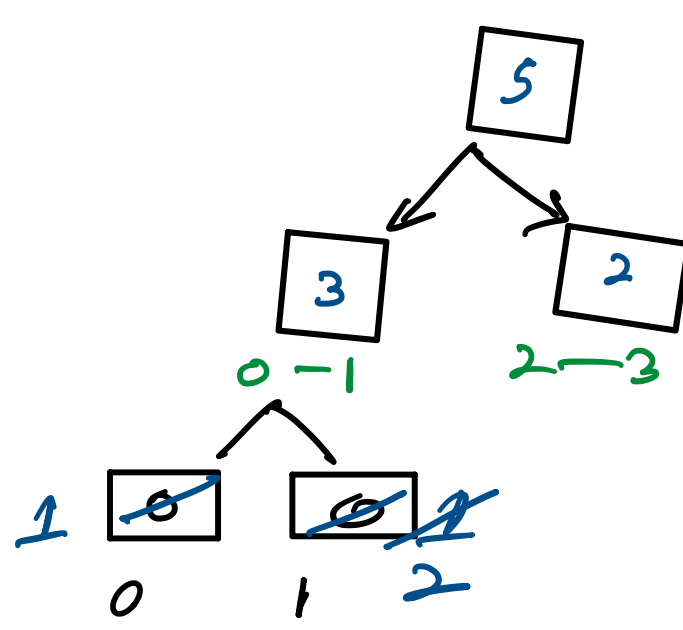
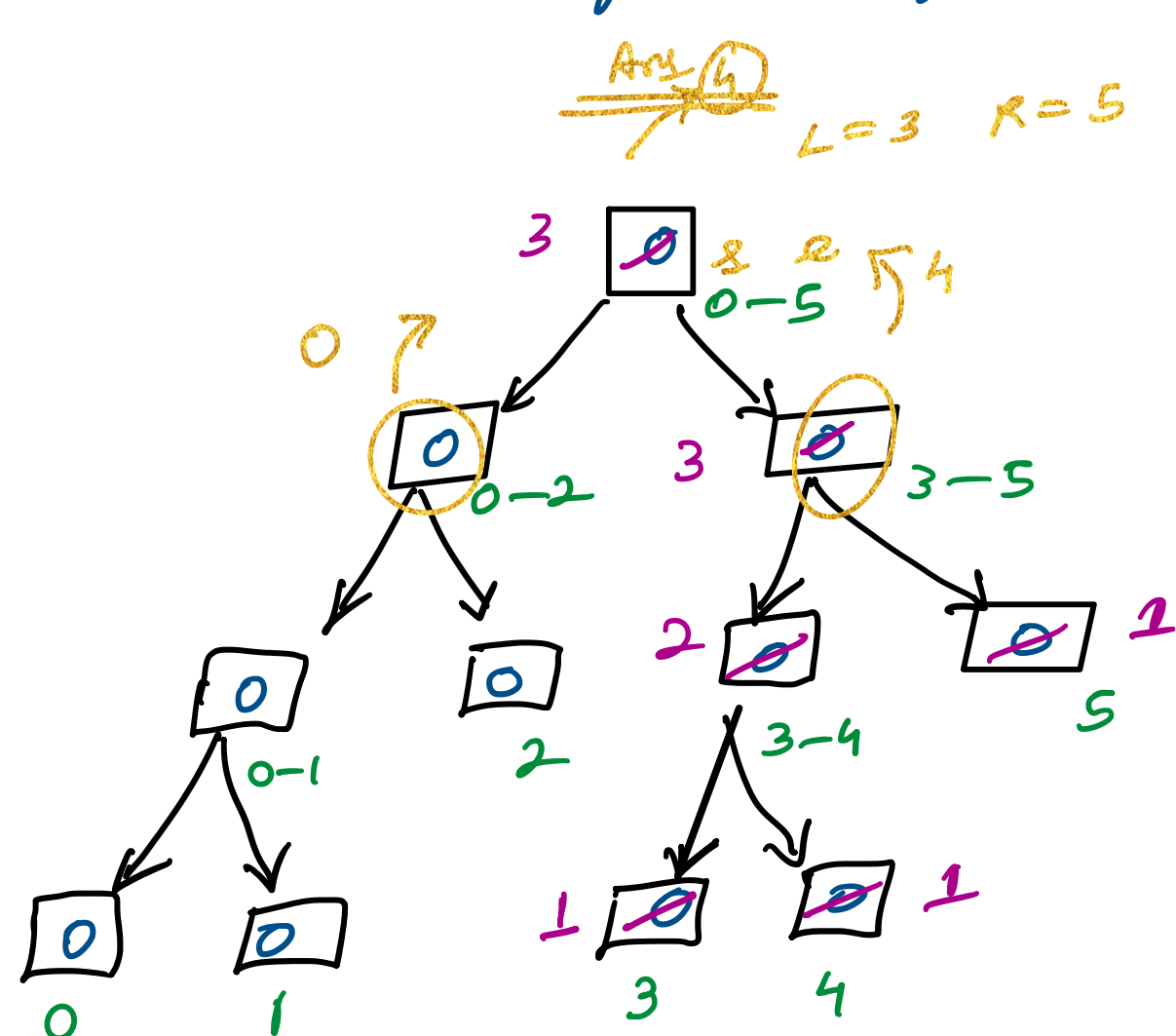


Q2 → Remove 1 bit in binary representation of $A[x]$.

Q3 → Count 1's in binary representation from index x to y.

Sum Segment Tree

Note data → count of 1's in binary representation from start to end.



A → [0, 0, 0, 0, 0, 0]

1 3 3 3 5
1 4 2 3
1 3
1 5

Q2 → Binary Update

Given an integer array st. $\forall i A[i] = 1$

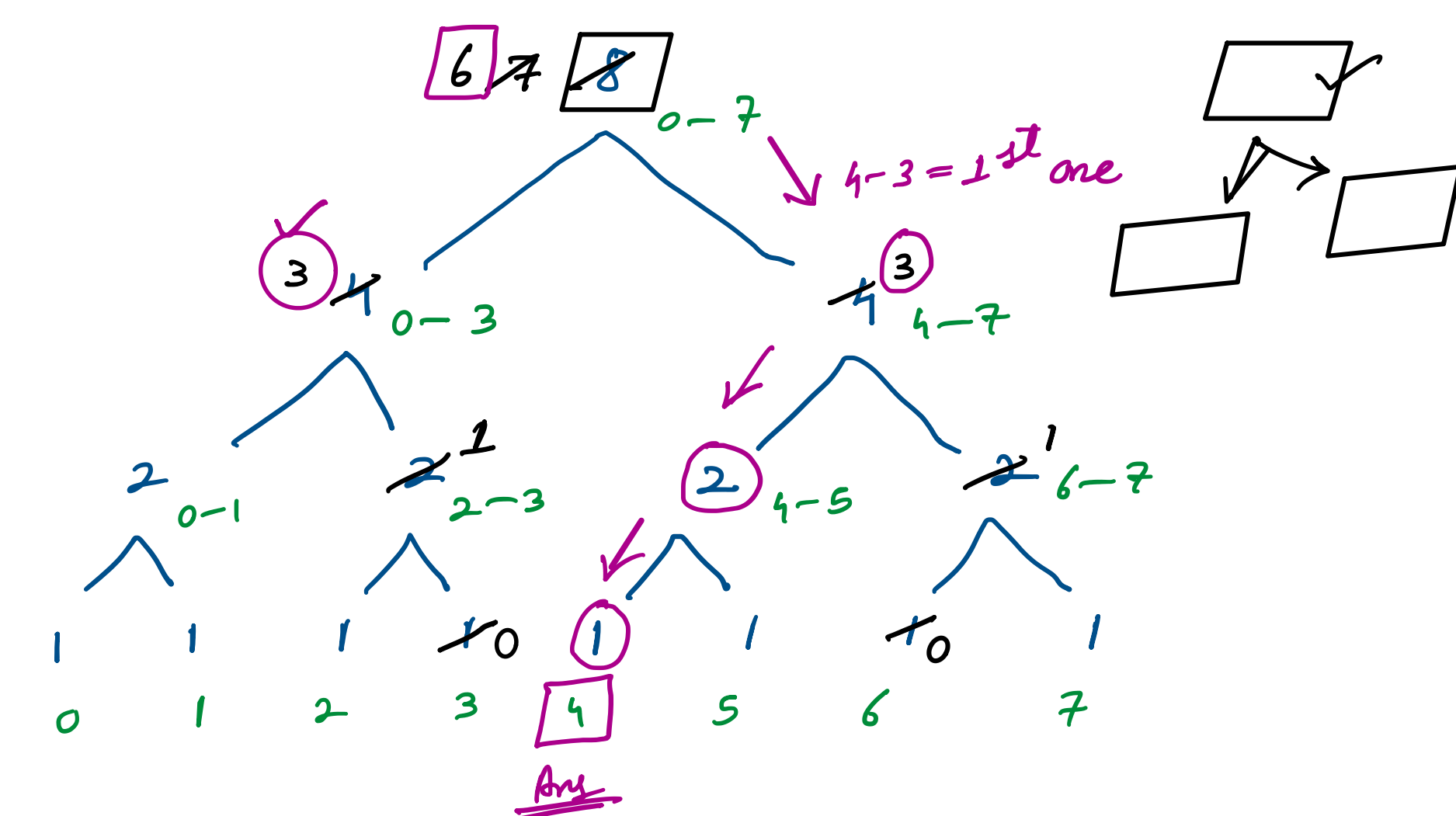
- Queries
- 1 x → $A[x] = 0$
 - 2 x → Find the index of xth 1 in the array.

A → [1, 1, 1, 1, 1, 1]

Brute → Q1 → $TC = O(1)$
Q2 → Traversal from start till we get ith 1. $TC = O(N)$

1 2
1 4
2 3 → Ans = 3
2 4 → Ans = 5

start → 0 end → Ans



2 4 → Find index of 4th one.

if (tree[le] >= x) idx = le;
else { idx = re; x = x - tree[le]; }

Q2 → Max Sum Query

A → [-8, 3, -1, 2, -9, 15, -6, 3]

Q2 → Find max sum among all subarrays?

Ans → Kadane's Algo

P → [0, 0, 3, 2, 4, 0, 15, 9, 12]

$P[i] = P[i-1] + A[i]$
if ($P[i] < 0$) $P[i] = 0$

- Queries
- 1 i x → $A[i] = x$
 - 2 l r → Find max sum subarray in range l to r.

A → [-8, 3, -1, 2, -9, 15, -6, 3]

1 5 -15
2 1 4

Ans = 3 - 1 + 2 = 4 (Ans)

16 16 10 10

0 1 2 3 4 5
3 8 -5 10 -12 6
P → 3 11 6 16 4 10
S → 10 7 -1 4 -6 6

11 11 6 6 10 10 6 4

3 8 -5 10 -12 6
3 11 6 16 4 10

- ans[idx] = max(ans[le], ans[re], maxPre[re] + maxSuf[le]); - (i)
- maxPre[idx] = max(maxPre[le], sum[le] + maxPre[re]); - (ii)
- maxSuf[idx] = max(maxSuf[re], sum[re] + maxSuf[le]); - (iii)
- sum[idx] = sum[le] + sum[re]; - (iv)

$TC = O(Q * \log(N))$

$SC = O(N)$