29/03/2024

Page No.
Date

Day 22 of DSA

Tasks

Check Box	Segment Tree
☑	Segment Tree Construction (Theory)
☑	- Implement Segment Tree Construction
☑	Query in segment
☑	Min/Max Element queries / Updates

Page No.
Date

* Segment Tree Construction

Given Array :

5	3	2	4	1	8	6	10
0	1	2	3	4	5	6	7

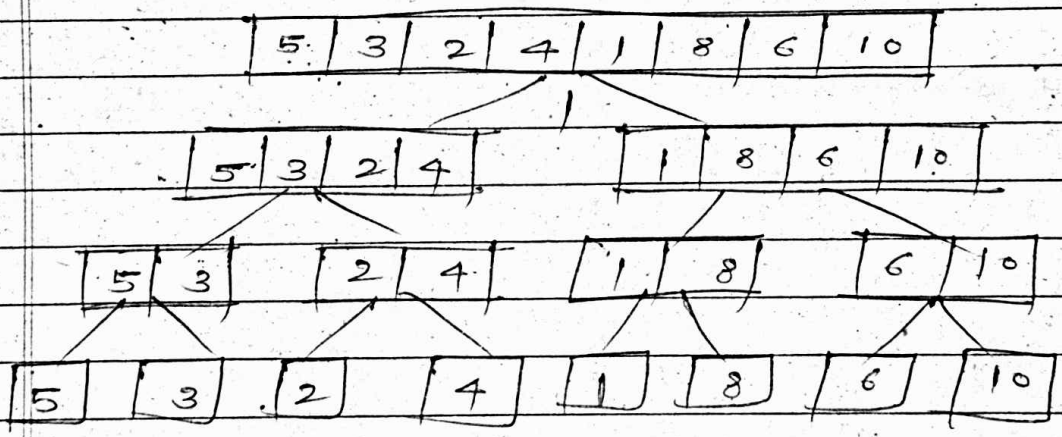
Power of Number 2 in Programming

→ Binary Representation of numbers.

All operations be it sum / subtraction / product, all are accomplished in O(1).

→ Division of Array (Divide and Conquer)

* We can divide the above array as :



Number of Nodes $= n + \frac{n}{2} + \frac{n}{4} + \dots 2 + 1$

$= 2$

For calculating sum,
we have to calculate No. of Terms

$$a r^{x-1} = n$$

$$a = 1, \quad r = 2$$

$$2^{(x-1)} = n$$

$$\log_2 (2)^{x-1} = \log_2 (n)$$

$$\boxed{x = 1 + \log_2 n}$$

$\therefore$ Number of levels $= 1 + \log_2 n$.

2 - Number of Nodes $= 1 + 2 + 4 + \ldots n$

$$= \frac{1 [(2)^{1+\log n} - 1]}{2-1}$$

$$= 2 [(2)^{1+\log n} - 1]$$

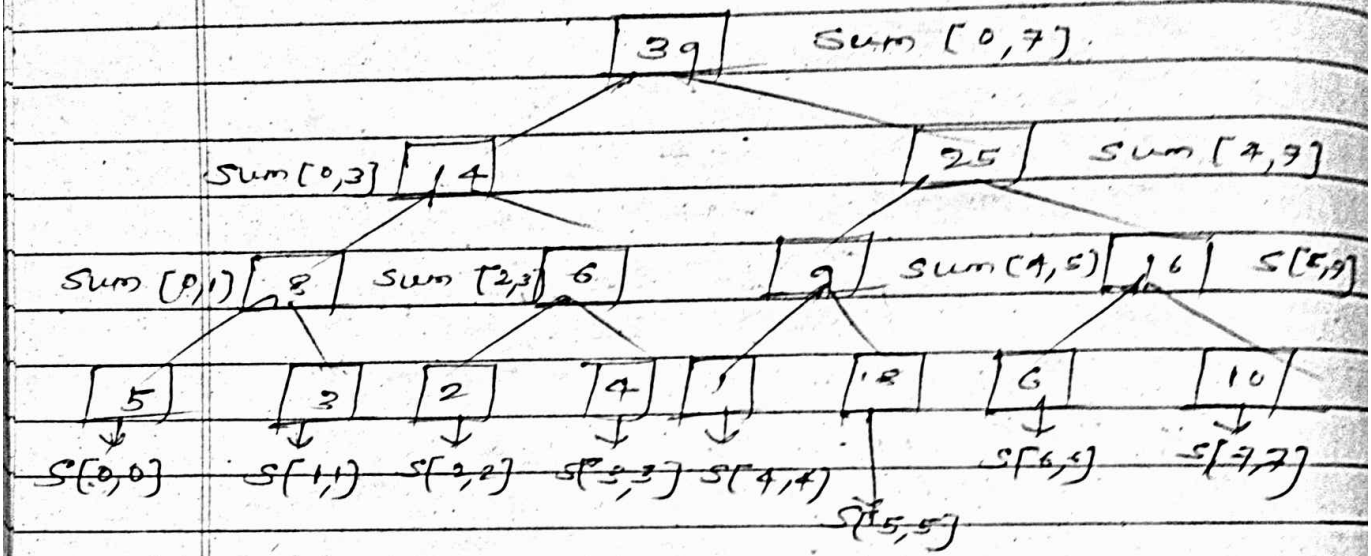
$$= (2n - 1)$$

For odd No:- Tree size $= 4 \times n$

Page No.
Date

Structure of segment Tree

Construction



Note:- We use divide & conquer to build segment tree

Pseudocode:

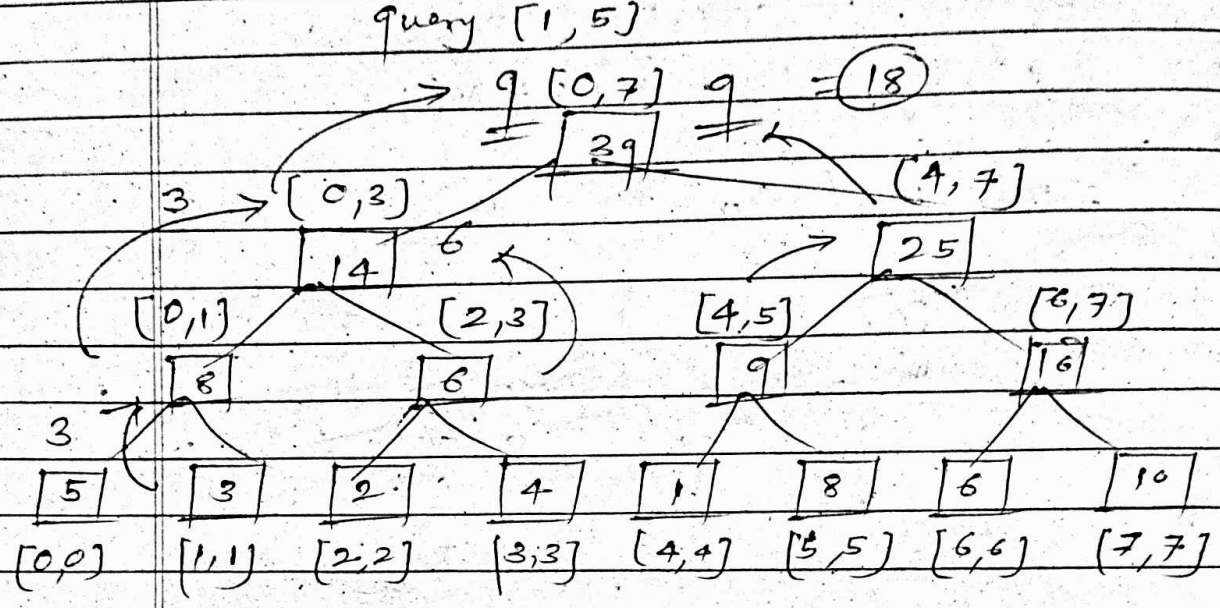
10^5+2

```
const int N = 1e5+2;
int a[N]; tree[4*N];

void build (int node, int st, int en)
{
    if(st==en)
    {
        tree[node] = a[st];
        return;
    }

    int mid = (st+en)/2;

    build(2*node, st, mid);
    build(2*node+1, mid+1, en);

    tree[node] = tree[2*node] +
                 tree[2*node+1];
```

Calling function

```
build(1, 0, n-1);
```

* Query in segment Tree



① Complete segment
② partial segment

* Pseudocode

```
int query (int node, int st, int en, int l, int r)
{
    if ( st > r || en < l) st en l r st en
        return 0;
```

② ✓

```
l st en r    if( l <= st && en <= r)
                return tree[node];

    int mid = (st+en)/2;

    int q1 = query (2*node, st, mid, l, r);
    int q2 = query ( 2* node, mid+1, en, l, r);

    return q1+q2;
}
```

* Maximum & Minimum Queries / updates

(1) Query = Output the maximum/minimum for the sub-array a[i...j]

(2) Update = Update the ith element of the array arr(idx) = updated value