15. Segment Trees

28 July 2024 13:58

⇒ Q/. Given an array and Q queries in the form of (l_i, r_i). Find subarray sum from $[l_i, r_i]$ for each query.

$a = [1, 2, 3, 4, 5, 6, 7]$

$[2, 3]$ → 7 $\quad [1, 4]$ → 14 $\quad (0, 3)$ → 10

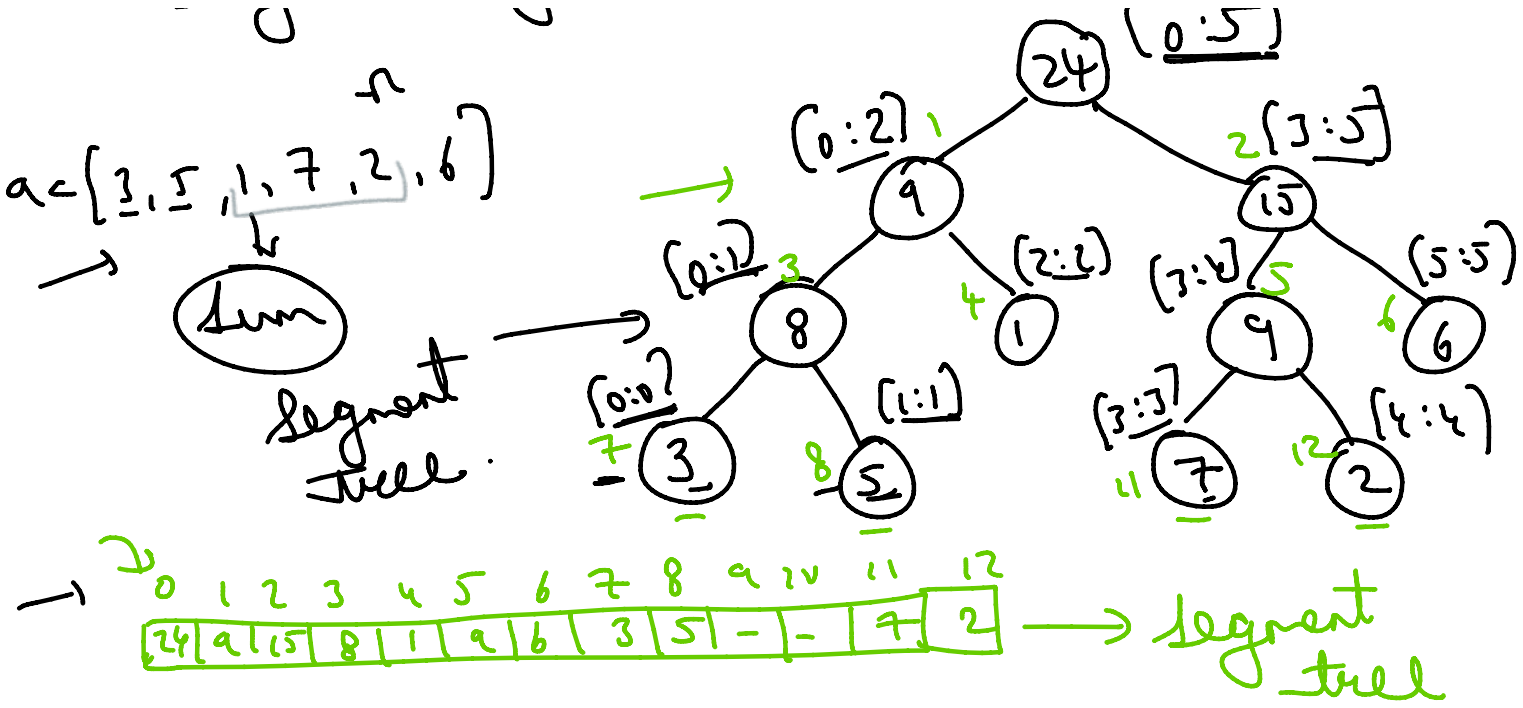
⇒ There are 2 types of queries

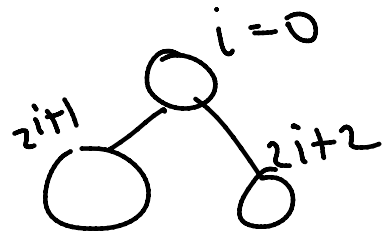
$[1, i, x]$ $\quad [2, l, r]$ → segment trees

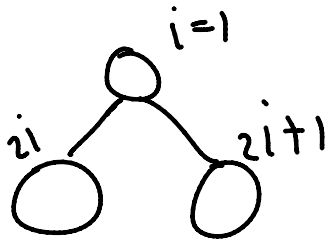
$[1, 3, 5]$

- Segment trees are used to solve range queries efficiently.
- Segment tree stores the information of segments of an array.
- It is a binary tree implemented using array.

(0, 0, 5) → 0 (24) [0:5]







$TC = O(n)$

a → n ←

seg → 4n ←

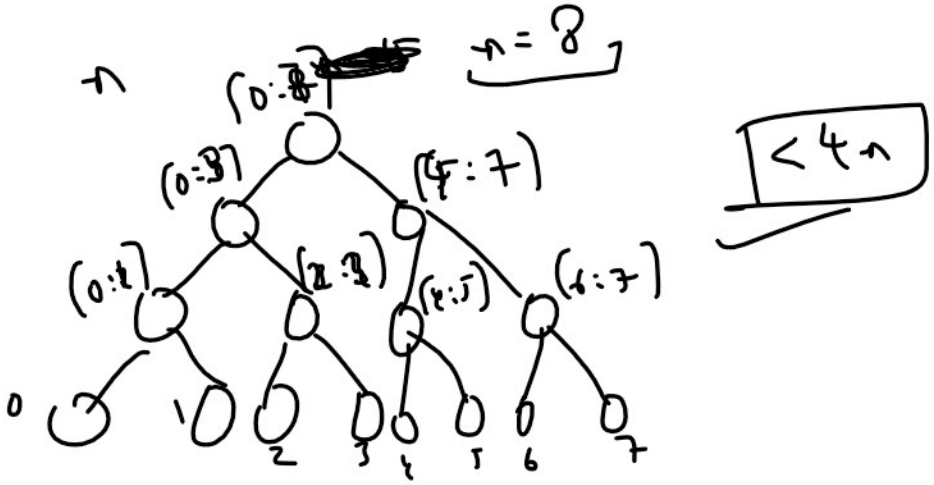
```
main(){
  build(0, 0, n-1, a, seg);
}

void build(int node, int l, int r, a[], seg[])
{
    if(l < r){
        int m = (l+r)/2;
        build(2*node+1, l, m, a, seg);
        build(2*node+2, m+1, r, a, seg);
        seg[node] = seg[2*node+1] + seg[2*node+2];
```

build(2*node+2, m+1, r); seg[node] = seg[2*node+1] + seg[2*node+2];

} else if(l==r){

seg[node] = a[l];

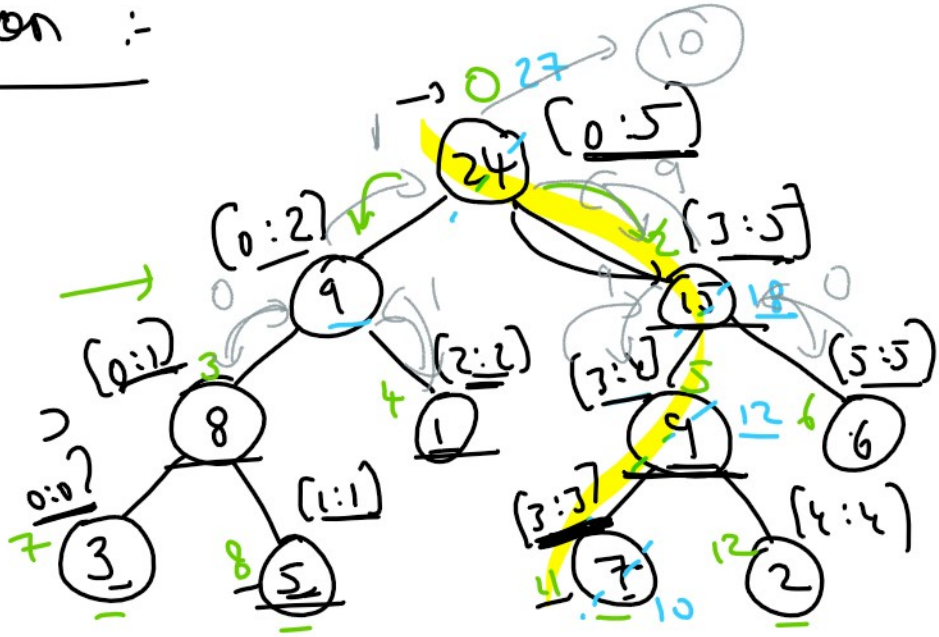
}



⇒ update function :-

(3, 10)

(2:4)



main() {

a[i] = x;

update(0, 0, n-1, i, x, seg);

}

void update(node, l, r, i, x, seg[]) {

if(l < r) {

m = (l+r)/2;

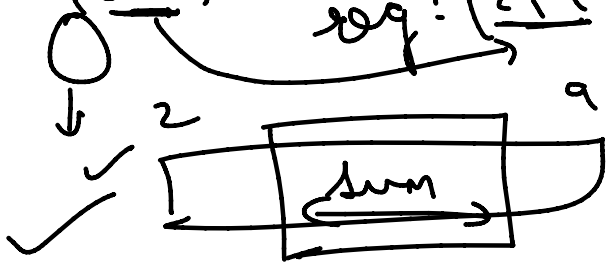
```
if(l<r){
    int m=(l+r)/2;
    if(i<=m)
        update(2*node+1,l,m,i,x,seg);
    else update(2*node+2,m+1,h,i,x,seg);
    seg[node]=seg[2*node+1]+seg[2*node+2];
}else{
    seg[node]=x;
}
}
```

TC = O(log n)

⇒ query function :- (l, r)

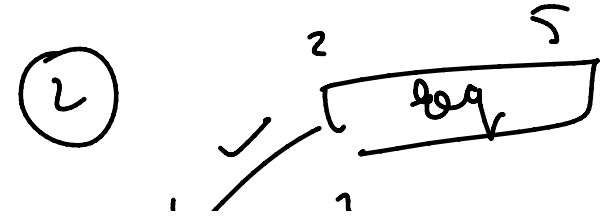
```
main(){
    query(0, 0, n-1, l, r, seg);
}
```

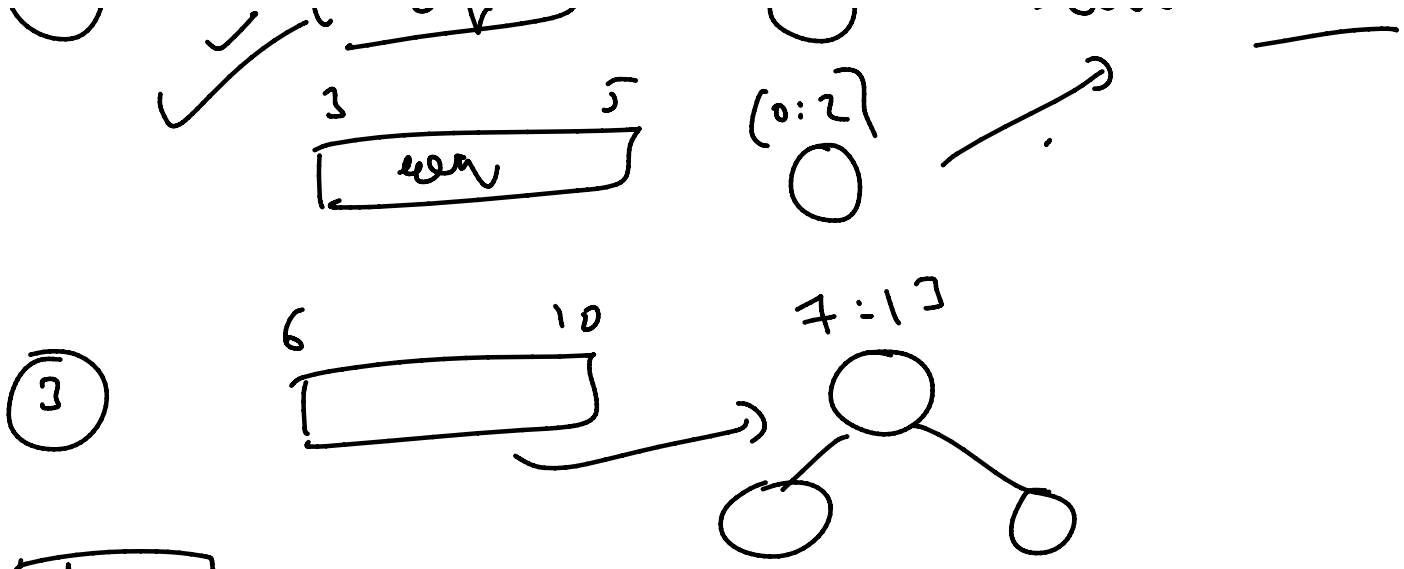
① (3:7) → seg:(2,9) return seg[node];



(2:10) (6:8)

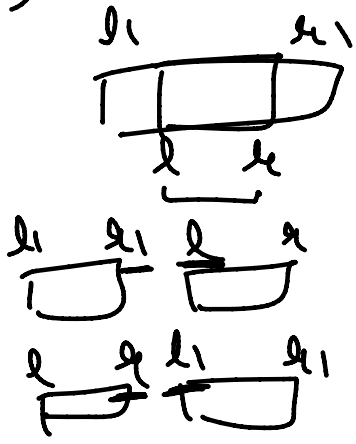
② (7:9) → return invalid





Sum

```
int query(node, l, r, l1, r1, seg[])
{
    if(l1<=l && r1>=r)
        return seg[node];

    if(l>r1 || l1>r)
        return 0;

    int m=(l+r)/2;
    int p1= query(2*node+1, l, m, l1, r1, seg);
    int p2= query(2*node+2, m+1, h, l1, r1, seg);
    return p1+p2;
}
```



TC = O(log n)

	Prefix	ST
build	O(n)	O(n)
x update	O(n)	O(log n)

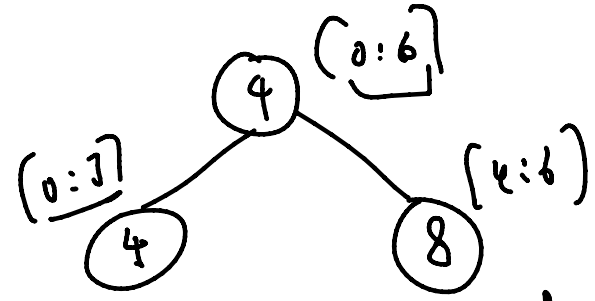
update	O(n)	O(log n)
query	O(1)	O(log n)

$a = [7, 8, 9, 1, 3, 5, 7, 2]$

q → 3 → [2, 1, 6] → 33

→ [1, 4, 20] →

→ [2, 1, 6] → 50



gcd(4, 8)

[4x1 4x2 4x3 4x4 8y1 8y2 8y3]

```
seg[node] =
    gcd(seg[2*node+1],
        seg[2*node+2]);
```

Euclid GCD fn :-

x > y → assumption

```
int gcd(x, y)
{   if(x % y == 0)
        return y;
    else return gcd(y, x % y);
}
```

}

else return gcd(y, x%y);

gcd(35, 15) → 5

↓

gcd(15, 5)

Q a, b → lcm(a, b) = $\frac{a \times b}{gcd(a,b)}$ →

⇒ **Lazy propagation**

⇒ Dealing with primality :-

Q/. n → prime ?

```
f=0;
for i=2 to n-1:
    if(n%i==0)
        f=1;
        break;
```

→ O(n).

→ n → $\frac{a \times b}{}$

a ≤ b

$a \le \sqrt{n}$

$b \ge \sqrt{n}$

n = 20

a = 4

axx

x = 5

$i \geq \sqrt{n}$

→ $O(\sqrt{n})$.

```
f=0;
for(i=2; i*i<=n; i++)
{
   if(n%i==0)
   { f=1; break;
   }
}
```

⇒ Any number can have at max one prime divisor $> \sqrt{n}$.

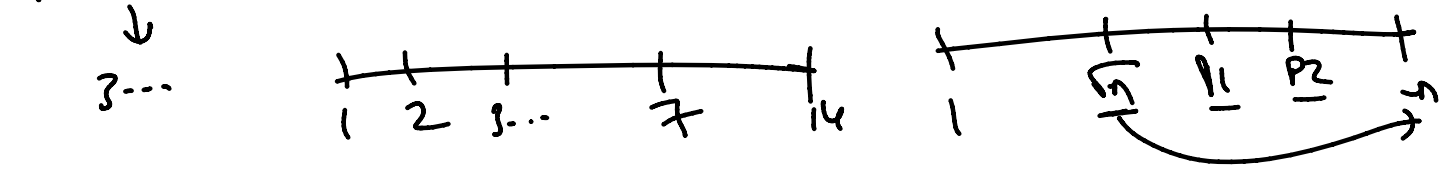
$n = p_1^{a_1} p_2^{a_2} p_3^{a_3} \cdots p_k^{a_k}$

$n = 200 \rightarrow 2 \times 100$

$\rightarrow 2 \times 50 \times 2$

$\rightarrow 2 \times 2 \times 2 \times 5^2 \rightarrow 2^3 \times 5^2$

$n = 14 \rightarrow 2 \times 7$



⇒ Sieve of Eratosthenes :-

↓

Used to find all prime numbers in a given range [1, n].

prime numbers efficiently from [1, n].

p(10^6 + 1); → initialize all values as 1.

p[0] = p[1] = 0;

```
for(i = 2 ; i*i <= 10^6 ; i++)
{
    if(p[i] == 1){
        for(j = i*i ; j <= 10^6 ; j += i)
        {
            p[j] = 0;
        }
    }
}
```

→ p[i] -> 1 → i → prime
p[i] -> 0 → i → composite.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1	1	+	1	+	1	+	+	+	1	+	1	+	+	+
			0		0		0	0	0		0		0	0	0

$TC = O(n \log\log n)$.

Q/ Count all unique prime divisors of n. $(n \le 10^{10})$.

⇒ $n = p_1^{q_1} p_2^{q_2} p_3^{q_3} p_4^{q_4} p_5^{q_5}$

$p_1 < p_2 < p_3 < p_4 < p_5$

=)

$n = p_1^{a_1} p_2^{a_2} p_3^{a_3} p_4^{a_4} p_5^{a_5}$

$i = 2$ to $\sqrt{n}$

$n = p_2^{a_2} p_3^{a_3} p_4^{a_4} p_5^{a_5}$

$p_1 < p_2 < p_3 < p_4 < p_5$

$2^3 \times 3^2 \rightarrow 72$

$n = 1800$

$= 2^3 \times 5^2 \times 3^2 \rightarrow 2^3 \times 3^2 \times 5^2 \rightarrow 1$

$i = 2, 3$

$3^2 \times 5^2 \times 13 \rightarrow 2$

$5^2 \times 13 \rightarrow 3$

$13 \rightarrow 4$

$200 \rightarrow 2^3 \times 5^2$

$5^2 \rightarrow 1$

2 to $\sqrt{200}$

$14 \ldots$

$n = 28$

$5 \ldots$

$2^2 \times 7$

7

$i = 2 \rightarrow c = 1$

$i = 3$

$n = 200 \rightarrow 2^3 \times 5^2 \rightarrow i = 2 \rightarrow c = 1$

$\leq \sqrt{n}$

$5^2 \rightarrow i = 3$

$i = 4$

$i = 5 \rightarrow c = 2$

1