

Segment Trees Lecture 2

Today's checklist



1. Range Updates via Lazy Propagation

Problems in updating a range



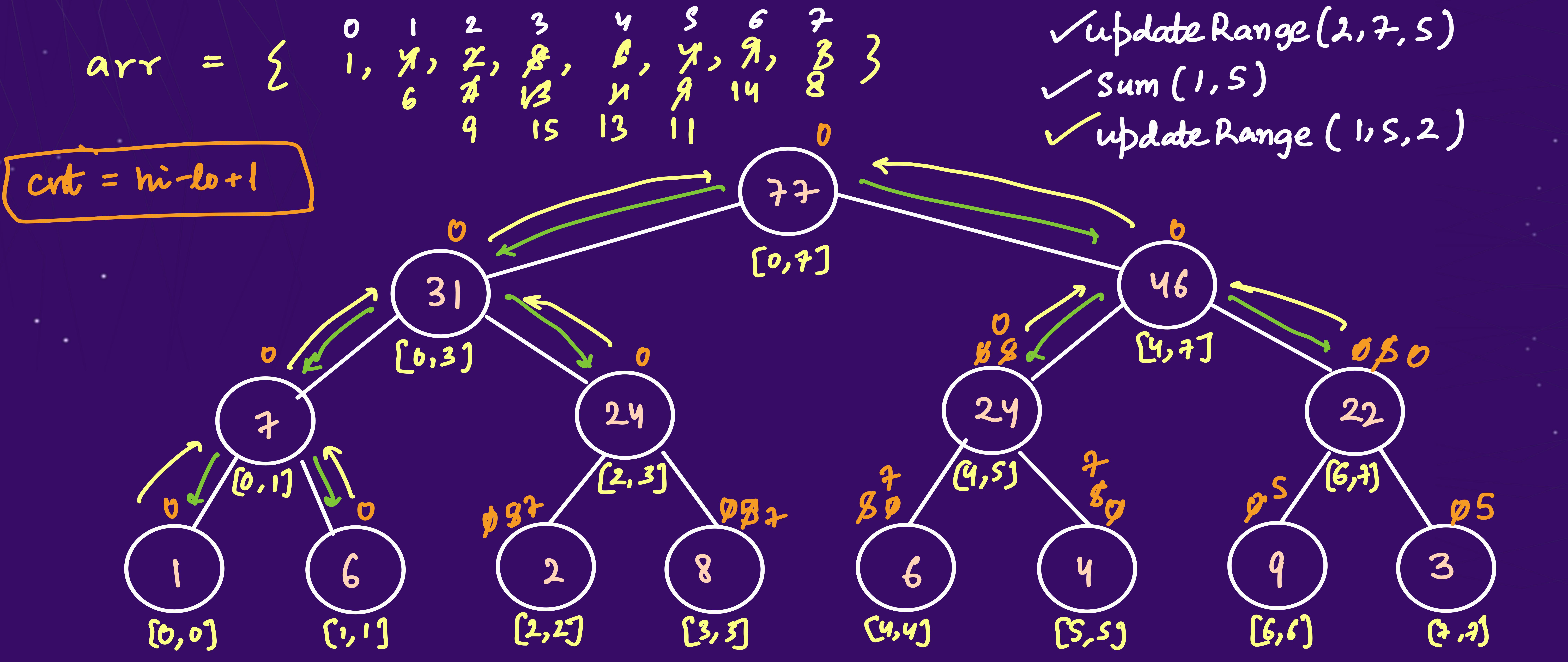
$$arr = \begin{cases} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1, & 4, & 2, & 8, & 6, & 4, & 9, & 3 \end{cases}$$

$$arr = \begin{cases} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1, & 4, & 5, & 11, & 9, & 12, & 9, & 3 & 3 \end{cases}$$

I 97 we update each index of range separately then
$$T \cdot C \cdot = O((r-l) \log n) \times O(n \log n)$$

Lazy Propagation





Sum
$$(1,3) = 14$$

update $(2,5,10)$
Sum $(1,3) = 4+12+18 = 34$



Lazy Propagation

```
int getSum(int i, int lo, int hi, int l, int r){
    // check for pending lazy updates
    if([lazy[i]]=0){
       int rangeSize = hi - lo + 1;
        st[i] += rangeSize*lazy[i];
        if(lo!=hi){ // send lazy to left and right child
            lazy[2*i+1] += lazy[i];
            lazy[2*i+2] += lazy[i];
        lazy[i] = 0;
   if(l>hi || r<lo) return 0;
    if(lo>=1 && hi<=r) return st[i];
    int mid = lo + (hi-lo)/2; // (lo+hi)/2
    int leftSum = getSum(2*i+1,lo,mid,l,r);
    int rightSum = getSum(2*i+2,mid+1,hi,l,r);
    return leftSum + rightSum;
```

```
void updateRange(int i, int lo, int hi, int l, int r, int val){
    // check for pending lazy updates
   if(lazy[i]!=0){
       int rangeSize = hi - lo + 1;
       st[i] += rangeSize*lazy[i];
       if(lo!=hi){ // send lazy to left and right child
            lazy[2*i+1] += lazy[i];
            lazy[2*i+2] += lazy[i];
        lazy[i] = 0;
   if(l>hi | r<lo) return;
   if(lo>=l && hi<=r){ // update entire [lo,hi]
       int rangeSize = hi - lo + 1;
       st[i] += rangeSize*val;
       if(lo!=hi){ // send lazy to left and right child
            lazy[2*i+1] += val;
            lazy[2*i+2] += val;
        return;
    int mid = lo + (hi-lo)/2; // (lo+hi)/2
   updateRange(2*i+1, lo, mid, l, r, val);
   updateRange(2*i+2,mid+1,hi,l,r,val);
   st[i] = st[2*i+1] + st[2*i+2];
```



JHANK YOU