Segment Tree

Concepts & Qns...

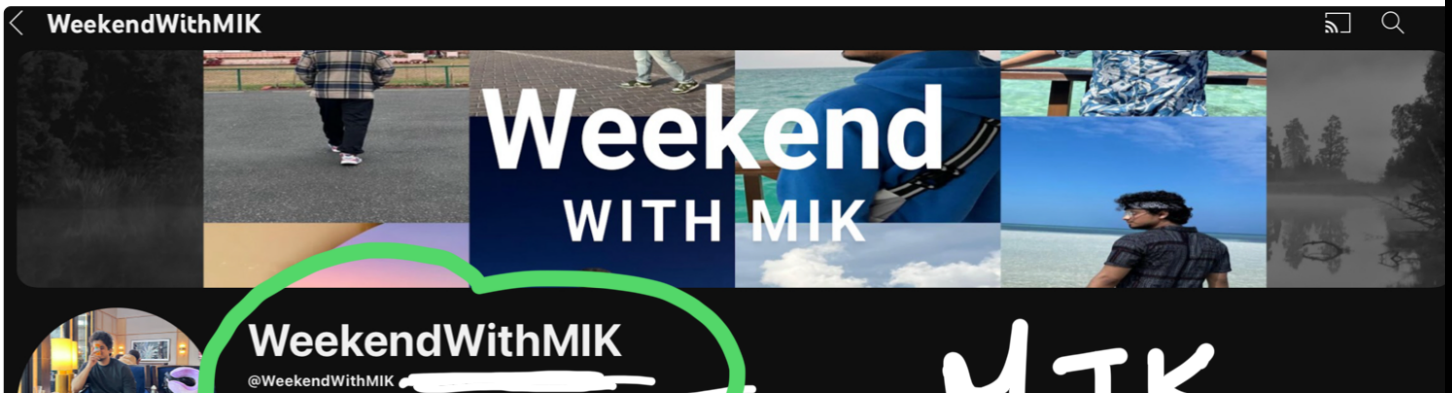
Facebook
Instagram → codestorywithMIK

(Twitter) → CSwithMIK

codestorywithMIK →

"No more fear of Segment Tree"

video - 9



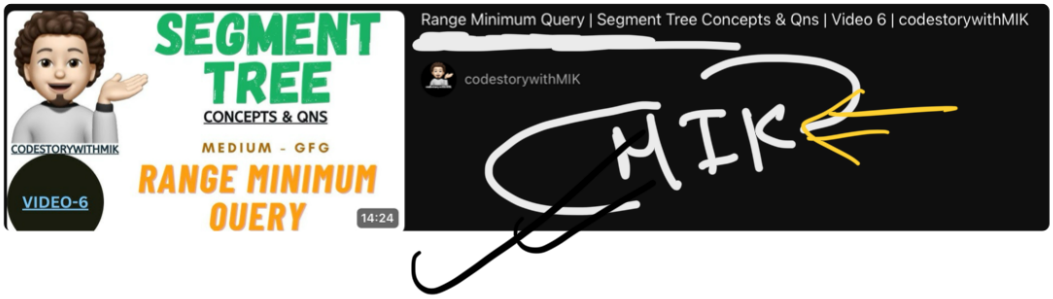
Try this channel to see my "Life behind the Scenes

Motivation :-

Shaayad aapke doston ka selection ho gaya ho, aur aapko lag raha ho that you're left behind...

Par yaad rakho, yeh race abhi khatam nahi hua hai.

Your time will come, aur jab aayega, sabhi dekhenge, and wo din "ee सबसे बड़ा दिन होगा" ...

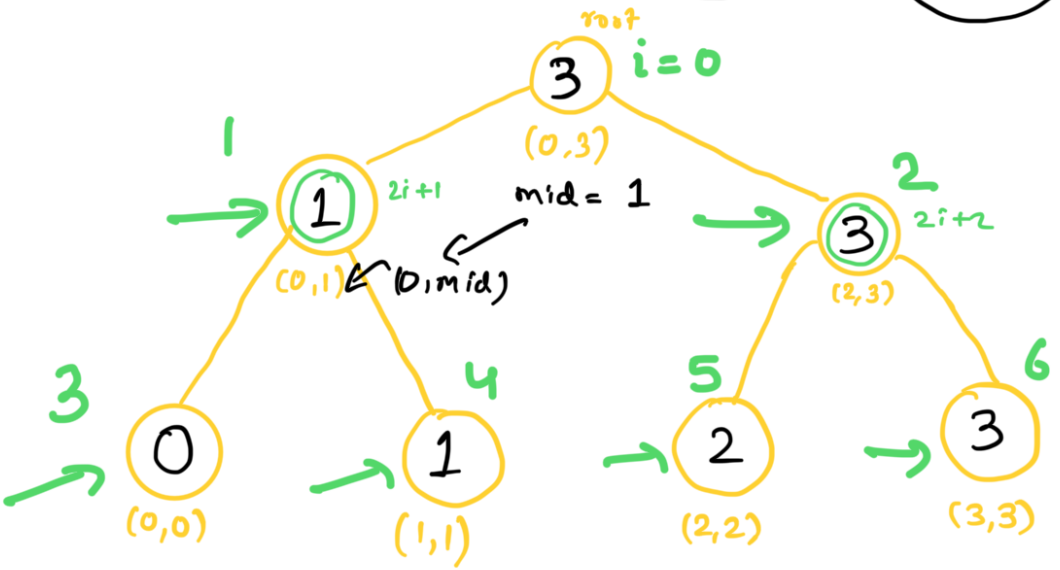


Range Max/Min

Index Query - PART-1

arr = { 1, 2, 3, 4 } → n = arr.size();

RMIQ = (a = 0, b = 2), (a = 2, b = 3), (a = 1, b = 3)



Range Maximum Query :- (RMQ)

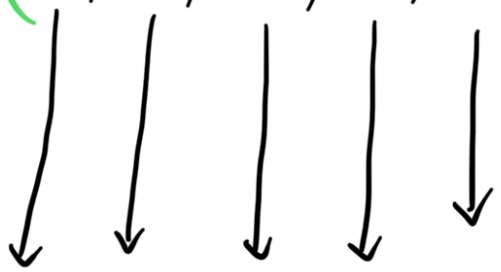
→ arr , n

→ int *segmentTree = new int[4*n];

⇒ buildSegmentTree (0, 0, n-1, segmentTree, arr);

```
void buildSegmentTree (int i, int l, int r, segTree, arr) {
    if (l == r) {                                   → storing index
        segment[i] = l;        // arr[l];
        return;
    }
    int mid = l + (r-l)/2;
⇒ buildSegmentTree (2*i+1, l, mid, segTree, arr);
⇒ buildSegmentTree (2*i+2, mid+1, r, segTree, arr);
    int leftMaxIdx = segTree[2i+1];
    int rightMaxIdx = segTree[2i+2];
    segTree[i] = max (segTree[2i+1], segTree[2i+2]);
    if (arr[leftMaxIdx] >= arr[rightMaxIdx]) {
        segmentTree[i] = leftMaxIdx;
    }
    else
        segmentTree[i] = rightMaxIdx;
}
```

```
querySegTree(a, b, 0, 0, n-1, segTree);
```



```
int querySegTree (start, end, i, l, r, segTree) {

    if (l > end || r < start) {
        return -1;          //invalid index
    }

    if ( l >= start && r <= end) {
        return segmenTree[i]; // returns index of
                              max element.
    }
```

```
int mid = l + (r-l)/2;

return max (querySegTree (start, end, 2*i+1, l, mid, segTree),
            querySegTree (start, end, 2*i+2, mid+1, r, segTree));
```

```
int leftMaxIdx = querySegTree (start, end, 2i+1, l, mid, segTree);

int rightMaxIdx = querySegTree (start, end, 2i+2, mid+1, r, segTree);
```

```
if (leftMaxIdx == -1) return rightMaxIdx;
if (rightMaxIdx == -1) return leftMaxIdx;
if (arr[leftMaxIdx] >= arr[rightMaxIdx])
    return leftMaxIdx;
return rightMaxIdx;
}
```

RMIQ.

T.C = log(n)

S.C = O(n).