Persistent Segment trees

Special class

Surya Kiran Adury • Jan 13, 2021

Surya Kiran Adury

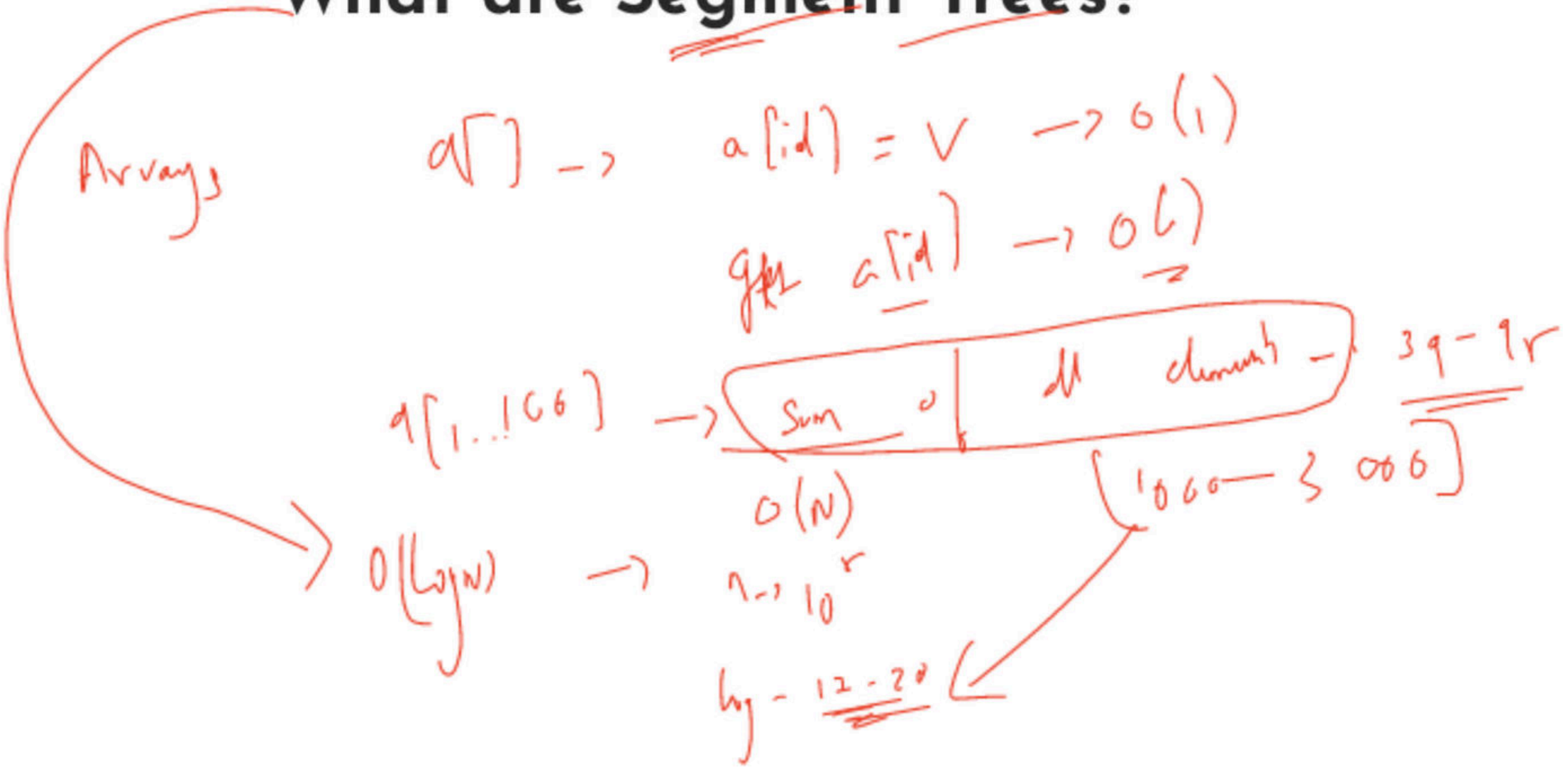
- ICPC WF - 2014, 15
- Google MTV 2017-20
- Google LON 2015-17
- B.Tech in ECE from IIT Roorkee


Question
What are we doing?

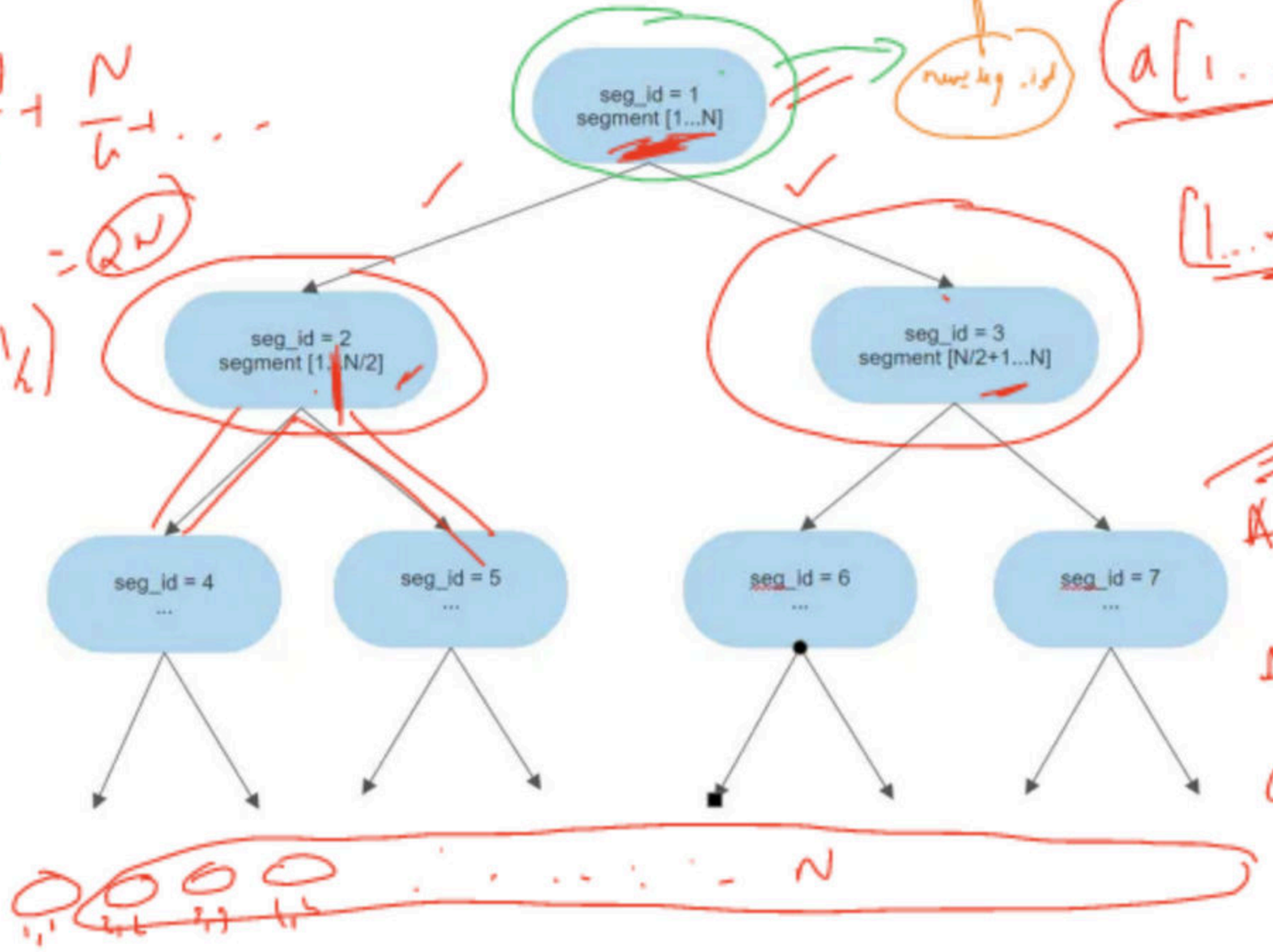
Objective

1. Class 1
   a. Fenwick Trees
2. Class 2
   a. Segment Trees
3. **Class 3**
   a. **Persistent Segment Trees**
4. Class 4
   a. Competitive programming problems

What are Segment Trees?

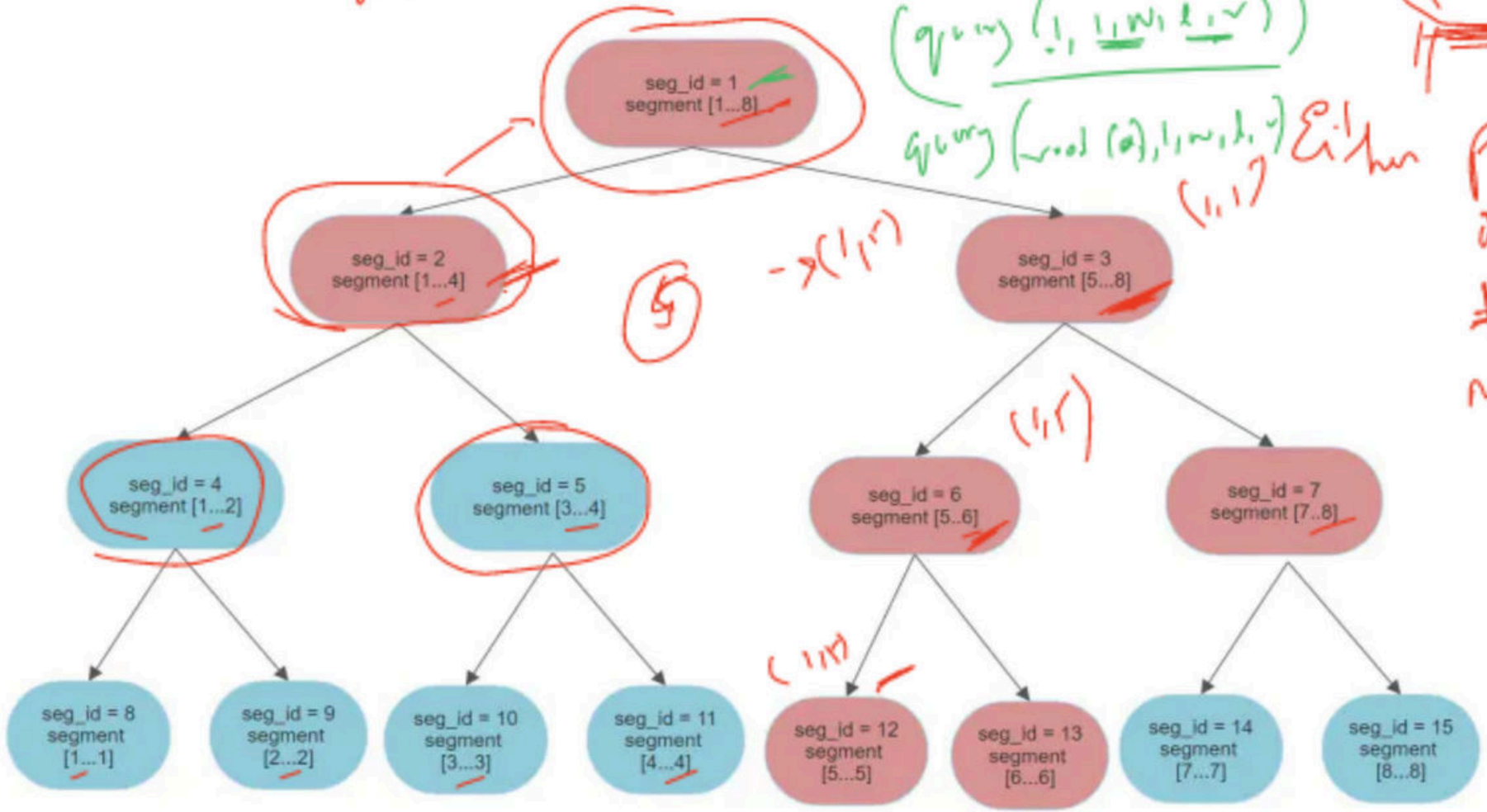


What are Segment Trees?

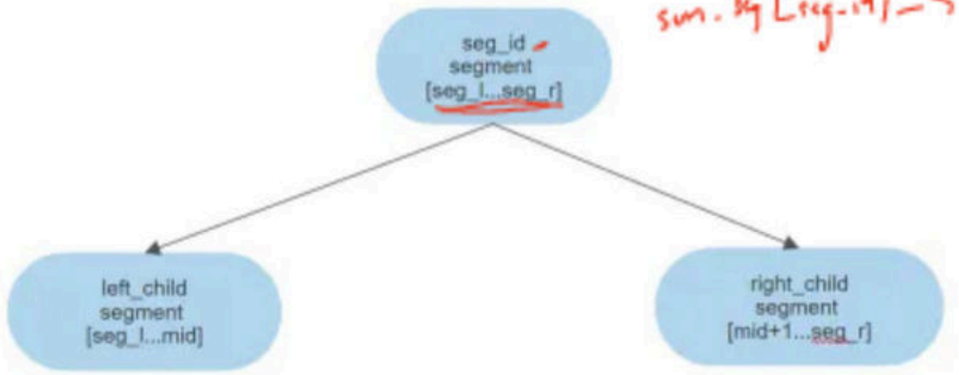


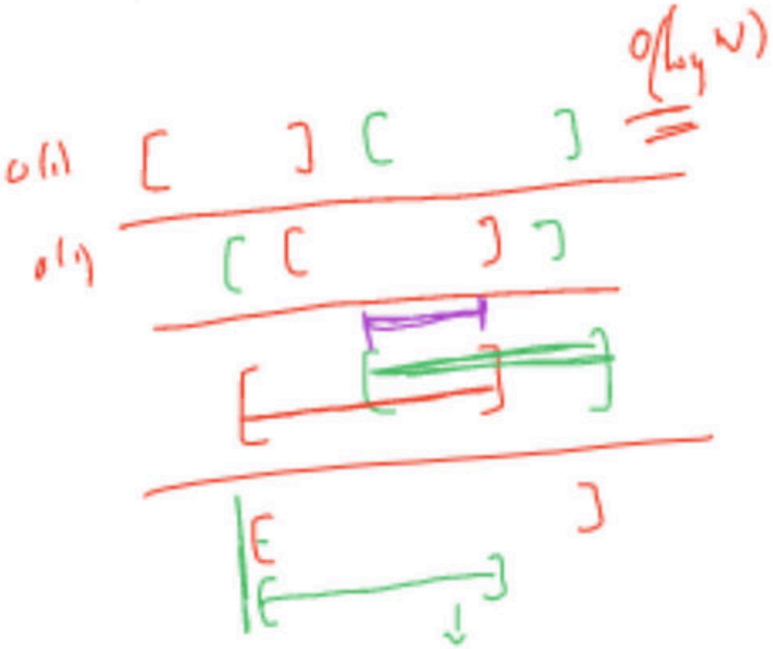
Query a Range

Query a Range



Query a Range





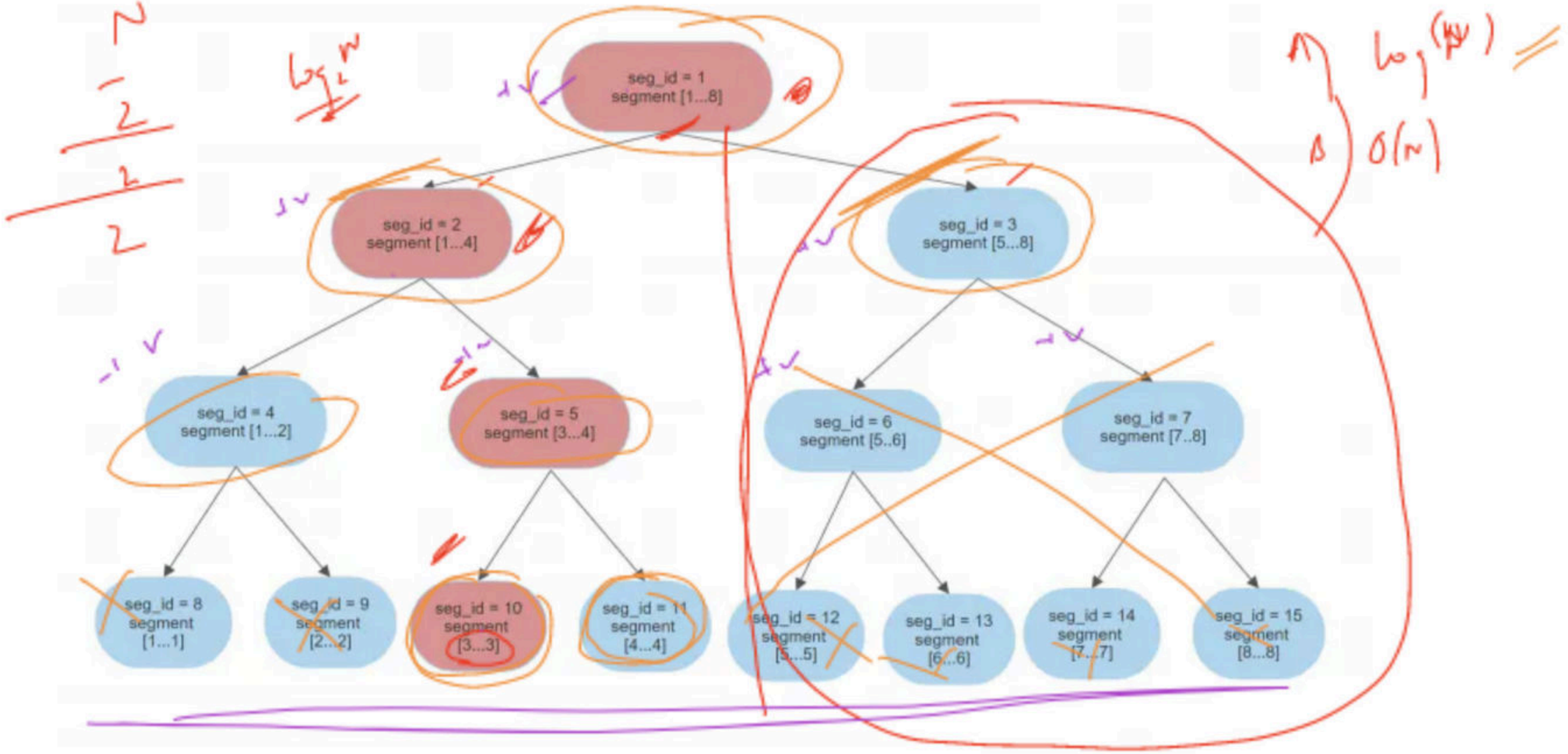
```
int sum_seg[1...S]; // sum_seg[seg_id] stores sum of the elements of
                    // the subarray a[seg_l, seg_r].
int a[N];           // array over which segment tree is built.

// Returns the sum of elements of overlap of the subarray a[query_l, query_r]
// and the subarray a[seg_l, seg_r].
int query(int seg_id, int seg_l, int seg_r, int query_l, int query_r) {
    if (query_l > seg_r || query_r < seg_l) {
        return 0; // No overlap between current seg_id and query.
    }
    if (quer_l <= seg_l && seg_r <= query_r) {
        return sum_seg[seg_id]; // Full overlap of current seg_id and query.
    }
    // We are left with partial overlap.
    // So we pass on to both children and return sum of values returned by them.
    int left_child = seg_id * 2;
    int left_l = seg_l;
    int left_ r = (seg_l + seg_r) / 2;

    int right_child = seg_id * 2 + 1;
    int right_l = left_r + 1;
    int right_r = seg_r;

    return query(left_child, left_l, left_r, query_l, query_r)
         + query(right_child, right_l, right_r, query_l, query_r);
}
```

Update an Index

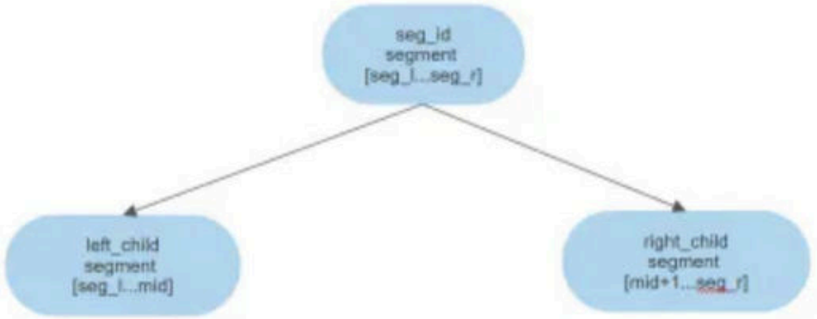
$A[1 \ldots N]$

$Seg[1 \ldots N]$

Update an Index



Update an Index




```
// Index update, adding "v" to a[id].
// Updates all segments to imitate adding "v" to a[id].
void index_update(int seg_id, int seg_l, int seg_r, int id, int v) {
    if (seg_l > id || seg_r < id) return; // No overlap.
    sum_seg[seg_id] += v;
    if (seg_l == seg_r) {
        return; // We are at leaf node.
    }
    int left_child = seg_id * 2;
    int left_l = seg_l;
    int left_ r = (seg_l + seg_r) / 2;

    int right_child = seg_id * 2 + 1;
    int right_l = left_r + 1;
    int right_r = seg_r;
    update(left_child, left_l, left_r, id, v);
    update(right_child, right_l, right_r, id, v);
}
```

unacademy | CODECHEF

2021 : The Year To QUIT PROCRASTINATING And LEARN CODING

Join Our Exclusive BATCHES

Pinnacle: Comprehensive and Concise Track to Become an Expert GOING LIVE ON 18TH JAN 2021

- Conquest 2021: Year Long Journey for Intermediate Coders to Become Experts (C++) - Live on 8th Jan 2021
- C++: Conquest 2021: From Programming Fundamentals to Career Readiness - Live on 8th Jan 2021
- Python: Conquest 2021: From Programming Fundamentals to Career Readiness - Live on 8th Jan 2021
- Java: Conquest 2021: From Programming Fundamentals to Career Readiness - Live on 8th Jan 2021

Resolve to become an expert level programmer in 2021 and subscribe at an expense even lesser than INR 90/ day

Learn Competitive Programming at Unacademy

PINNACLE Batch: Starting from 18th January 2021

Structured learning for intermediates to become expert level coders- Detailed Topic Coverage with Extensive Problem Solving

Instructors: Highly competent technical minds with **ICPC world finals, IOI medals**, IOI team training experience and **Codeforces Grandmasters** as accolades

Develop end to end subject matter expertise required to get placed in top product firms or create your own tech company or crack international coding contests

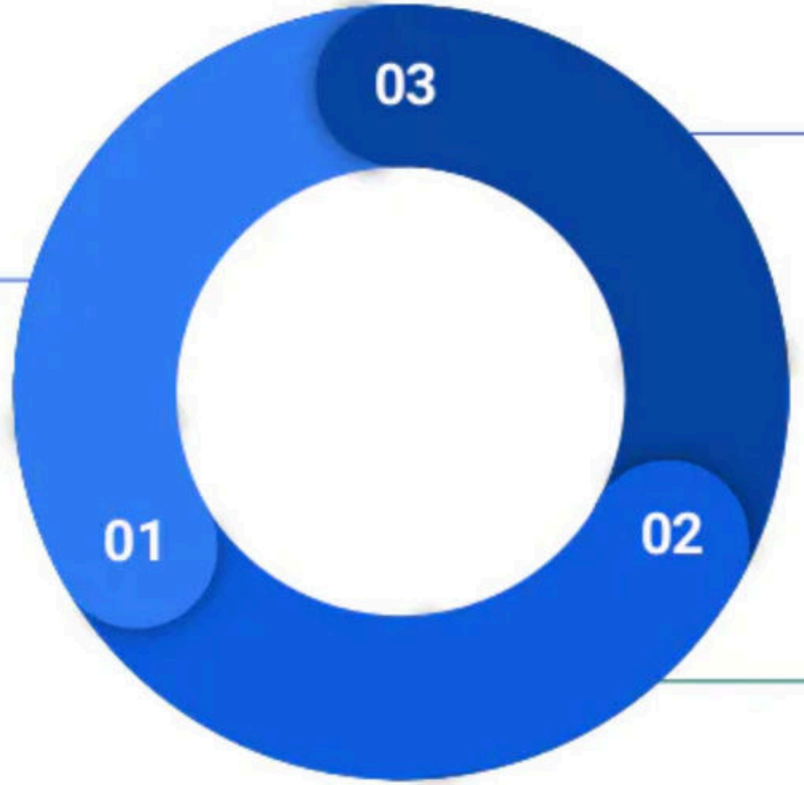
Industry accepted **Codechef Certification that comes free of cost along with the 1-year/ 6-month subscription,** upon successful course completion

The expense is even lesser than INR 90/ day with our 1 year subscription to avail all of it

What you will get

Live Interactive Classes

Attend live interactive classes with our top educators. Interact during class with educators to get all your doubts resolved



Doubt Support

If you get stuck in any problem post class- Get your doubts resolved by our expert panel of teaching assistants and community members instantly

Practice Relevant Problems @ CodeChef

Each class comes with a set of curated practice problems to help you apply the concepts in real time.

Topic-wise Structure: Pinnacle Batch

Week	Topic
Week 1	Sorting and Searching- Concepts and Problem Solving
Week 2-7	Data Structures- Concepts and Interview Problem Solving
Week 8-12	Additional Concept in C++
Week 13-14	Greedy Algorithms with Classical Problem Solving
Week 15-17	Data Structures 2 - Square Root Decomposition and Advanced Problems
Week 18-19	Number Theory and Interview Questions
Week 20-22	Recursion and DP Concepts and Handpicked Problem Solving
Week 23-27	Discrete Mathematics in C++- Concept to Problems
Week 28-32	Graph Algorithms- Advanced Problems
Week 33-35	Segment Trees
Week 36-38	Advanced Dynamic Programming
Week 39-41	Computational Geometry
Week 42-52	ICPC Regionals + World Finals Problem solving

One Subscription and Unlimited Access to All Batches/ Courses

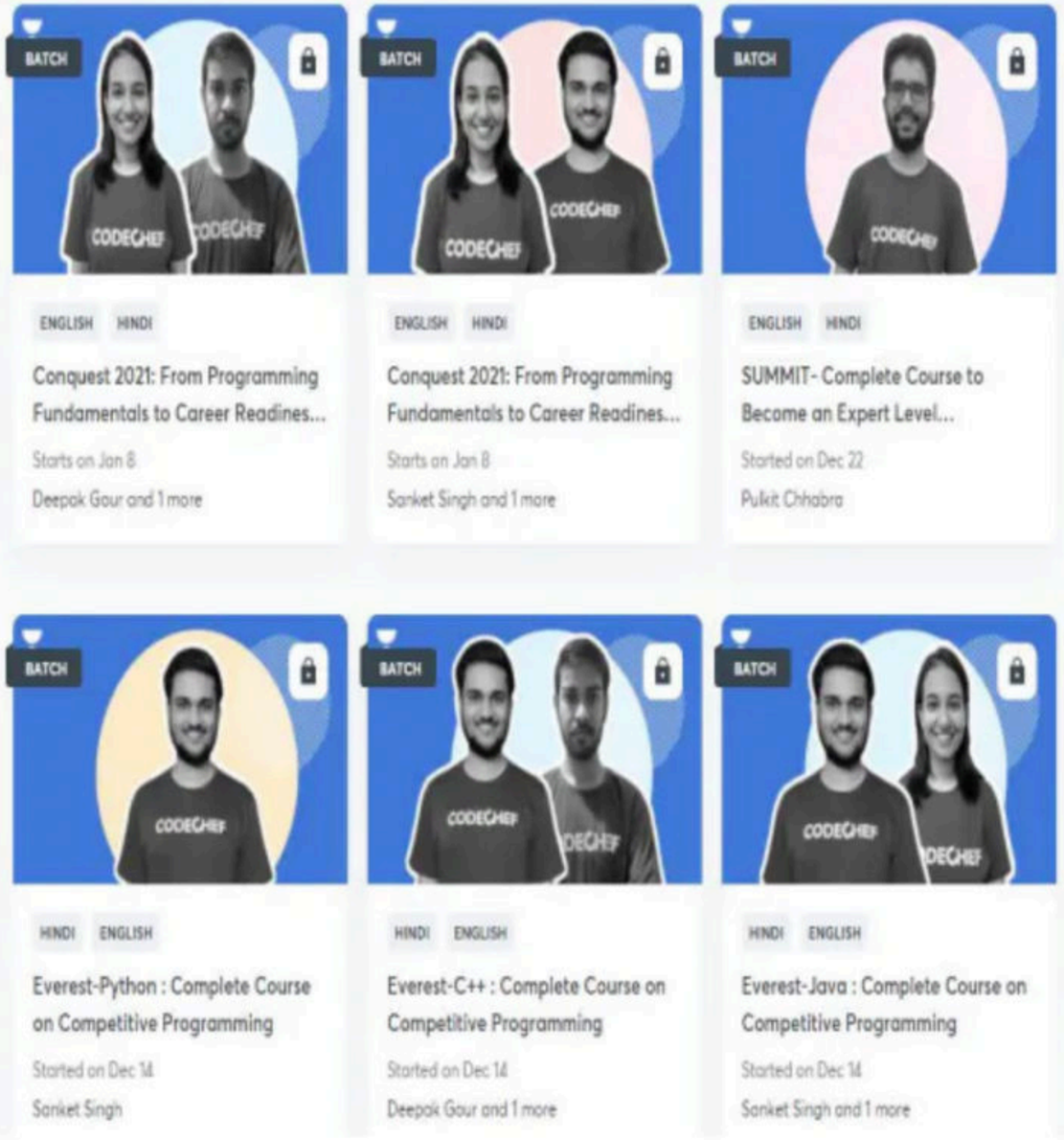
Batch Getting Live on 18th January 2021:

PINNACLE: Comprehensive and Concise Track to Become an Expert (C++)

Recently Live Batches

- Conquest 2021: From Programming Fundamentals to Career Readiness **(C++)**
- Conquest 2021: From Programming Fundamentals to Career Readiness **(Java)**
- Conquest 2021: From Programming Fundamentals to Career Readiness **(Python)**
- Conquest 2021: Year Long Journey for Intermediate Coders to Become Experts (C++)

And many more for all levels of programmers
Visit the Batches section in Unacademy



Educators

Tanuj Khattar

ACM ICPC World Finalist - 2017, 2018. Indian IOI Team Trainer 2016-2018. Worked @ Google, Facebook, HFT. Quantum Computing Enthusiast.

Sanket Singh

Software Development Engineer @ LinkedIn | Former SDE @ Interviewbit | Google Summer of Code 2019 @ Harvard University | Former Intern @ISRO

Pulkit Chhabra

Codeforces: 2246 | Codechef: 2416 | Former SDE Intern @CodeNation | Former Intern @HackerRank

Riya Bansal

Software Engineer at Flipkart | Former SDE and Instructor @ InterviewBit | Google Women TechMakers Scholar 2018

Triveni Mahatha

Qualified ICPC 2016 World Final. Won multiple Codechef Long Challenges (India). ICPC Onsite Regionals' Problem setter and Judge. IIT Kanpur.

Deepak Gour

ICPC World Finalist 2020 | Former Instructor @InterviewBit | Software Engineer at AppDynamics

Educators

Himanshu Singh

World Finalist ICPC 2020, Winner Techgig Code Gladiators 2020, Winner TCC '19, 2020 CSE Graduate from IIT BHU, Works at Nutanix

Nishchay Manwani

Hey I am Nishchay Manwani from CSE, IIT Guwahati and I'm a Seven star on Codechef and International Grandmaster on Codeforces.

Murugappan S

Software engineer at Google. Have won many programming contests. Max Rating of 2192 in codeforces and 2201 in codechef.

Vivek Chauhan

Codechef: 7 stars (2612) India Rank 6, Codeforces: MASTER (2279), Won Codechef Long Challenges(India), TCO20 Southern Asia Runner up

and many more joining soon...

Teaching Assistants support on chat and Doubts Forum

CODECHEF Discuss

You may face issue with markdown in posts. In such cases, report it here along with the post link.

unacademy Live Classes / CodeChef Practice & Doubts / CodeChef Doubt Forum

Clear your Doubts with our Expert Panel of Teaching Assistants & Community Members

Leave no room for doubts. Create a topic.

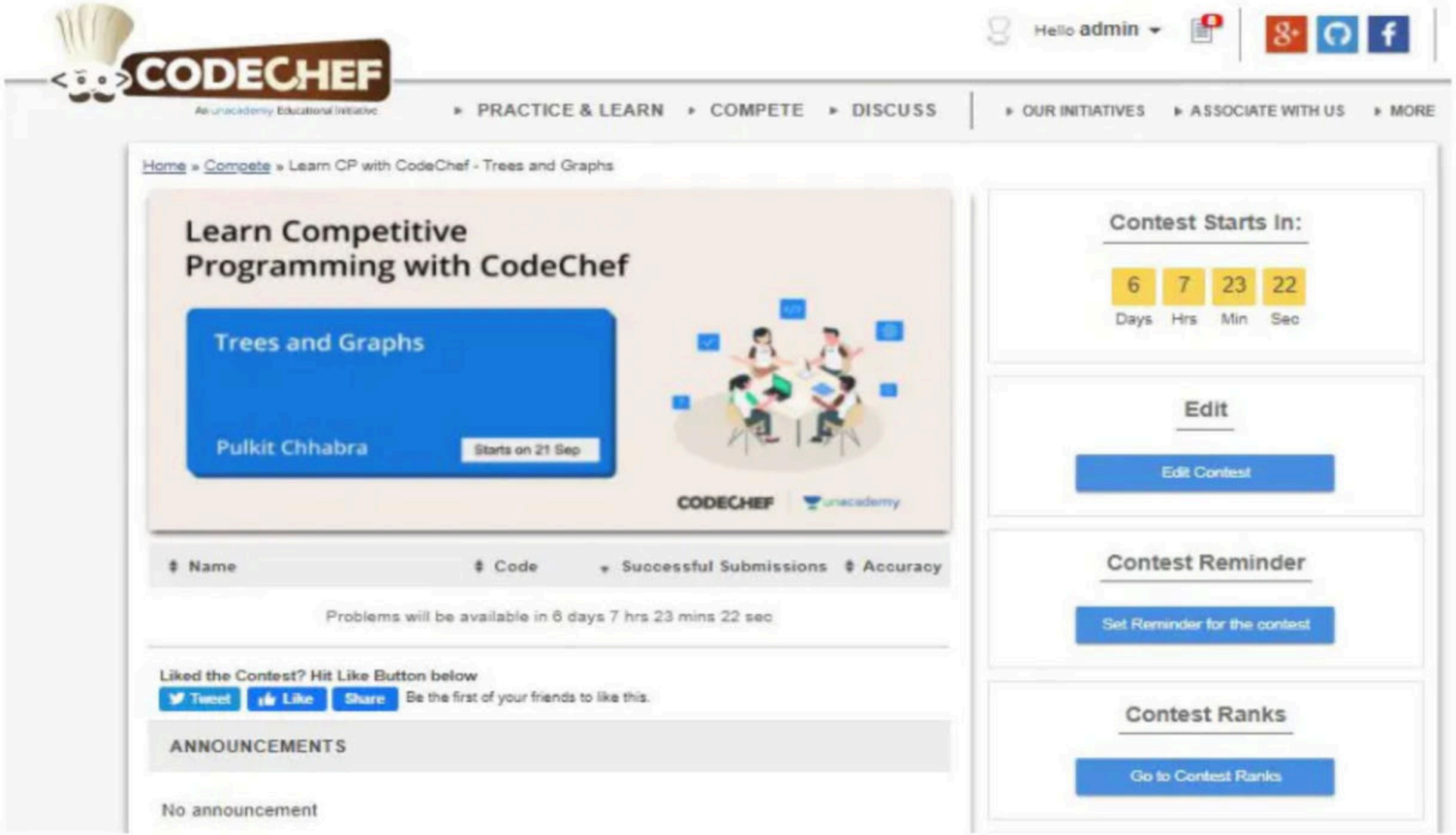
CODECHEF unacademy

Learn CP on Unacademy Plus | all tags | Latest | Top | Bookmarks | Edit | + New Topic

Topic	Replies	Views	Activity
About the Learn CP on Unacademy Plus category	1	6	2d

There are no more Learn CP on Unacademy Plus topics. Why not create a topic?

Course-wise Practice Problems



Flexible Subscription Plans

Competitive Programming subscription

Choose a plan and proceed

No cost EMI available on 6 months & above subscription plans

1 month		₹5,400 per month	₹5,400 Total (Incl. of all taxes)
3 months	11% OFF	₹4,800 per month	₹14,400 Total (Incl. of all taxes)
6 months	25% OFF	₹4,050 per month	₹24,300 Total (Incl. of all taxes)
12 months	54% OFF	₹2,475 per month	₹29,700 Total (Incl. of all taxes)

adurysk

Awesome! You got 10% off

Proceed to pay

<90 days

10%

adurysk

Proceed to pay

Segment trees → Query → Range

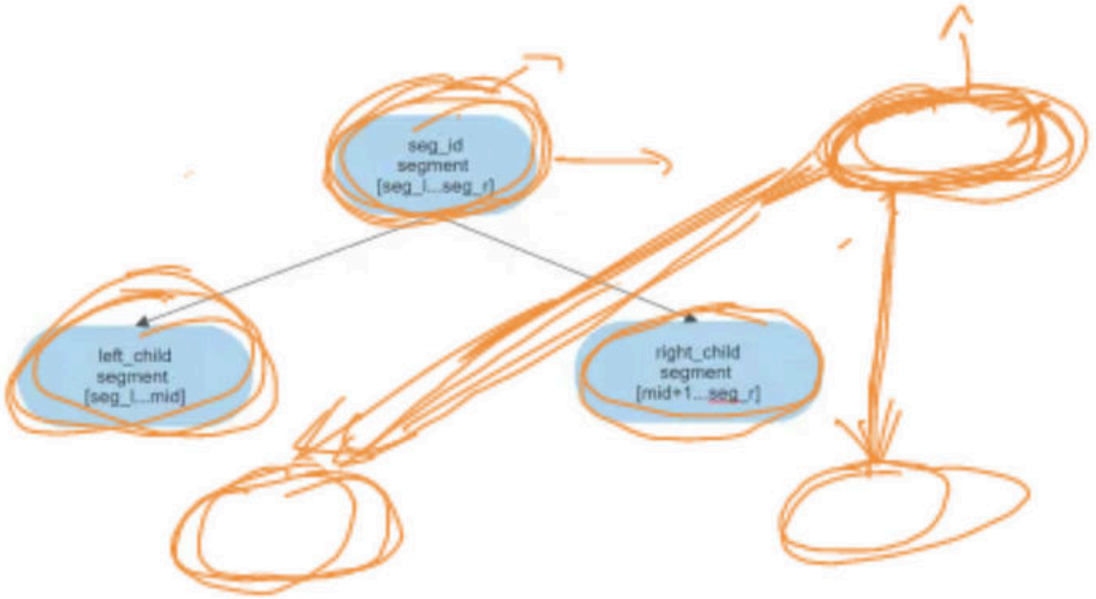
Updating → Index

What are persistent Segment Trees?

Range Minimum Query ⟶ (query_l, query_r)

Maximum

Updating an index



A) O(N)

B) O(log N)

```
int left_child[1...S];
int right_child[1...S];
int sum_seg[1...S];
int seg_cnt;

int copy(int seg_id) {
    seg_cnt++;
    sum_seg[seg_cnt] = sum_seg[seg_id];
    left_child[seg_cnt] = left_child[seg_id];
    right_child[seg_cnt] = right_child[seg_id];
    return seg_cnt;
}

int update(int seg_id, int seg_l, int seg_r, int id, int v) {
    int new_seg_id = copy(seg_id);

    if (id < seg_l || seg_r < id) return new_seg_id; // No overlap.

    seg_sum[new_seg_id] += v;
    if (seg_l == seg_r) return new_seg_id; // We are at leaf node.

    int left_l = seg_l;
    int left_r = (seg_l + seg_r) / 2;

    int right_l = left_r + 1;
    int right_r = seg_r;

    left_child[new_seg_id] = update(left_child[seg_id], left_l, left_r, id, v);
    right_child[new_seg_id] = update(right_child[seg_id], right_l, right_r, id, v);

    return new_seg_id;
}
```

Problem 1

A[1 . . . N]

Day 1

Day 2

Day 3

Day d

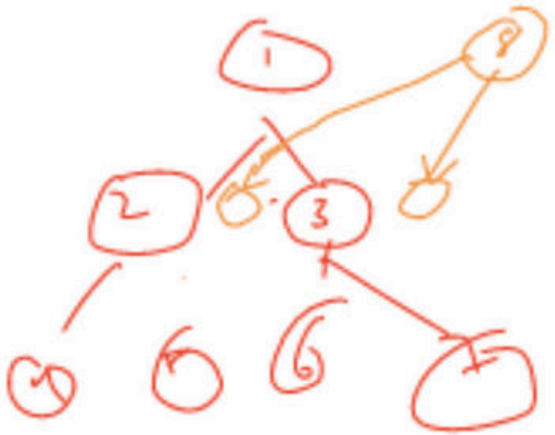
Day current day

operation → id, +v

(l, v), d (ok current day)

int root_id[1 . . . D]

root_id[1, 8, x . . .]



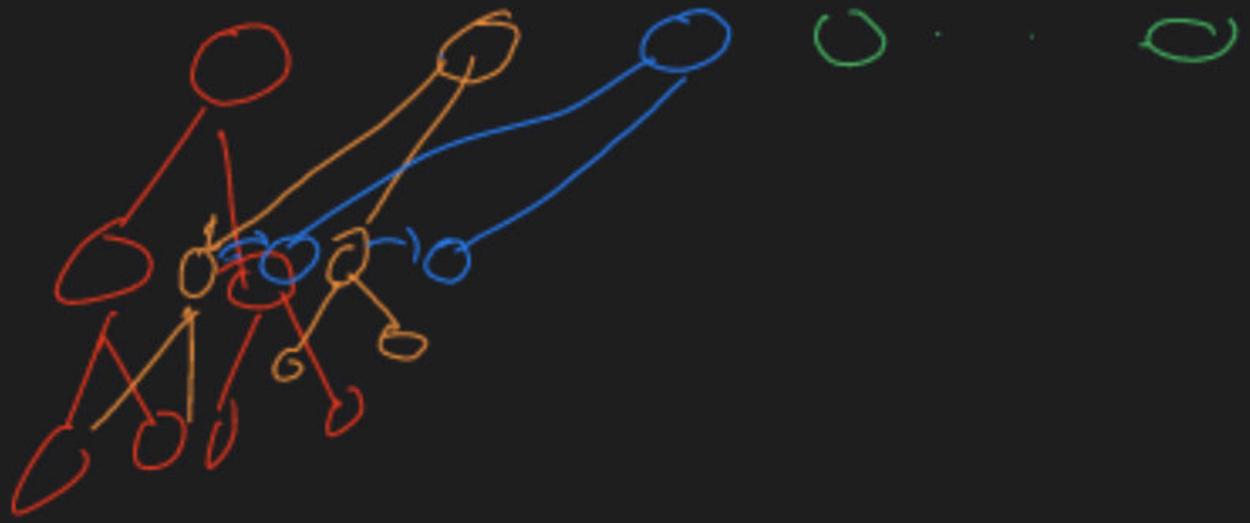
query (root_id[d], l, r, 1, N)

root_id(current day) = update (root_id[prev], 1, N, id, v)

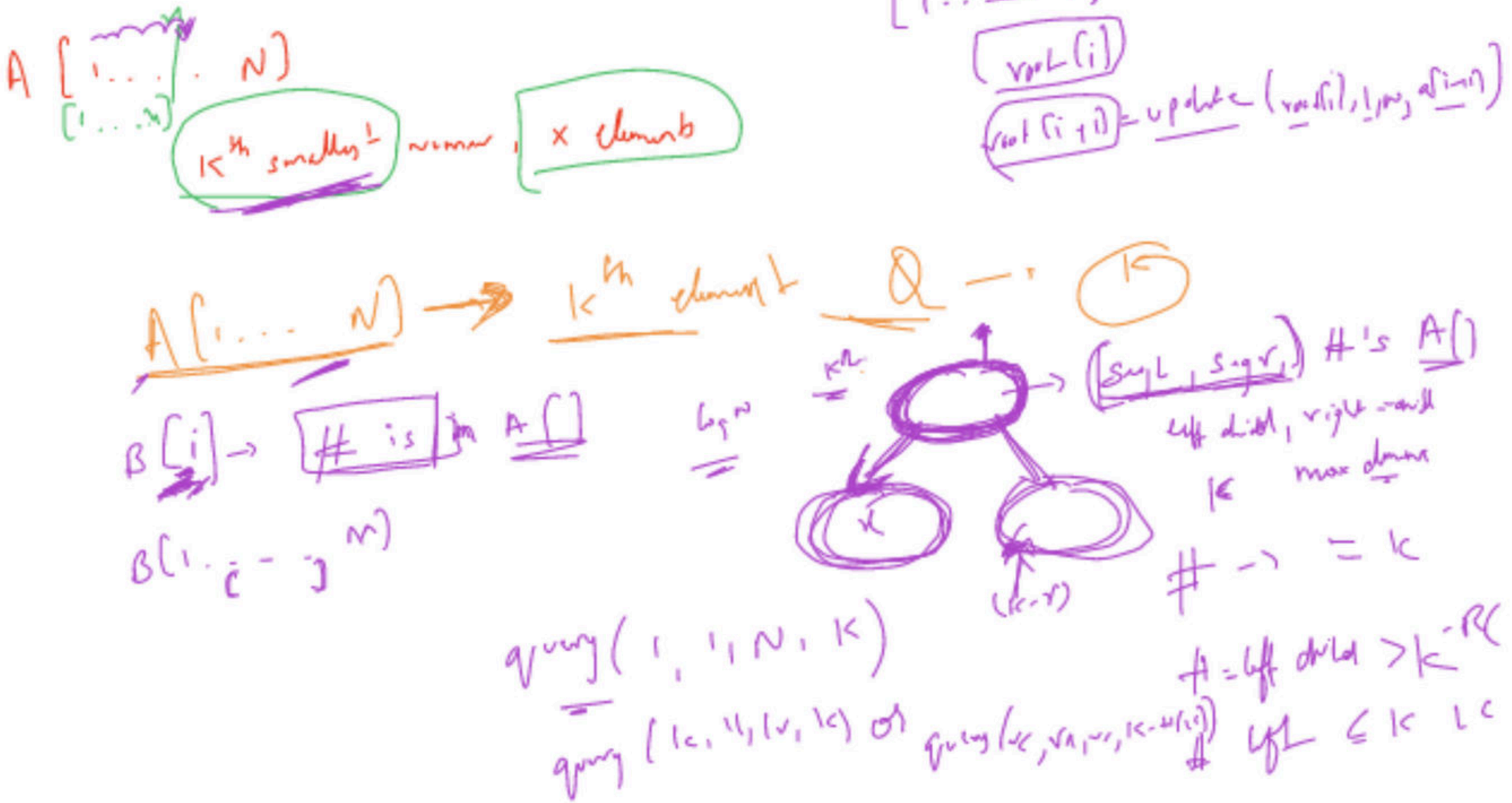
root_id[1 ... B]

root[current_day] = update(root[current_day-1], 1, N, id, v);

cout << query(root[d], 1, N, d, v) << endl;



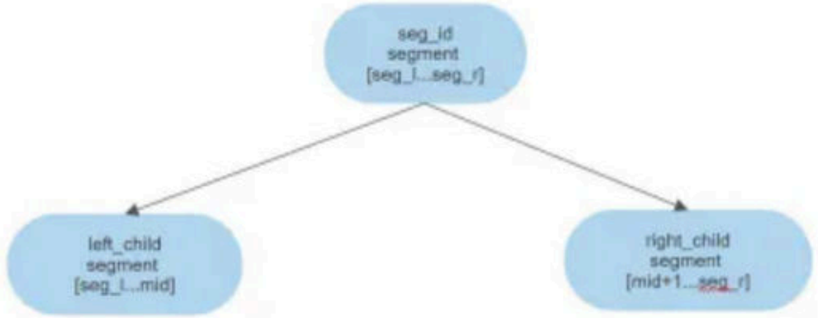
Problem 2


A [1 . . . N]
[1 . . . i]
root(i)
root(i+1) = update(root(i), ..., ...)
k^th smallest number
x elements
A[1 . . . N] → k^th element
Q →
B[i] → # is in A[]
B[1 . . . N]
log N
k
(sum, segs) # 's A[]
left child, right child
max element
# -> = k
# = left child > k - RC
# left ≤ k LC
query(1, 1, N, k)
query(lc, 1, lv, k)
query(lc, va, mv, k - u(lc))

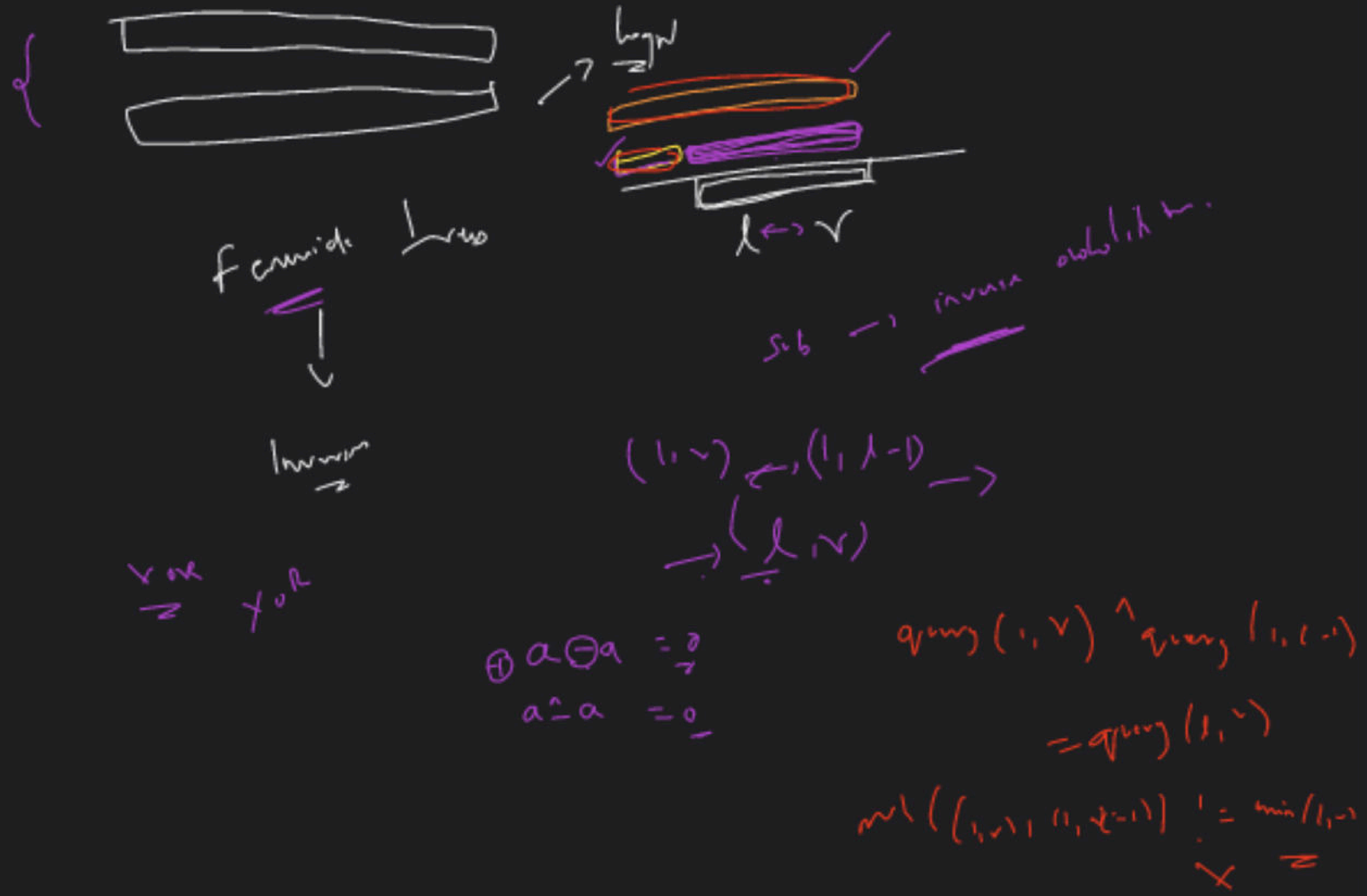
Problem 3

Update a Range of Segment Trees
Lazy propagation

Update a Range - Lazy propagation




```
// Range update, adding "v" to each element of sub_array a[id].
// Updates all segments to imitate adding "v" to a[id].
void range_update(int seg_id, int seg_l, int seg_r, int query_l, int query_r,
                  int v) {
    if (seg_l > id || seg_r < id) return; // No overlap.
    if (query_l <= seg_l && seg_r <= query_r) {
        seg_sum[seg_id] += v * (seg_r - seg_l + 1);
        seg_add[seg_id] += v; // This is for future use, to push to children.
        return;
    }
    int left_child = seg_id * 2;
    int left_l = seg_l;
    int left_ r = (seg_l + seg_r) / 2;

    int right_child = seg_id * 2 + 1;
    int right_l = left_r + 1;
    int right_r = seg_r;
    // Push seg_add[seg_id] to left child. This is O(1).
    range_update(left_child, left_l, left_r, left_l, left_r, seg_add[seg_id]);
    // Push seg_add[seg_id] to right child. This is O(1).
    range_update(right_child, right_l, right_r, right_l, right_r, seg_add[seg_id]);
    // Regular update to left child O(logn).
    range_update(left_child, left_l, left_r, query_l, query_r, v);
    // Regular update to left child O(logn).
    range_update(right_child, right_l, right_r, query_l, query_r, v);

    // New seg_sum can be got from information of children.
    seg_sum[seg_id] = seg_sum[left_child] + seg_sum[right_child];
    // set seg_add[seg_id] back to zero since previous updates were already
    // passed on to children.
    seg_add[seg_id] = 0;
}
```



logN
Fenwick Tree
Inversion
XOR
XOR
a^a = 0
query (1, r) ^ query (1, l-1)
= query (l, r)