Solving competitive programming problems using Segment trees

Special class

Surya Kiran Adury • Jan 14, 2021

Surya Kiran Adury

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

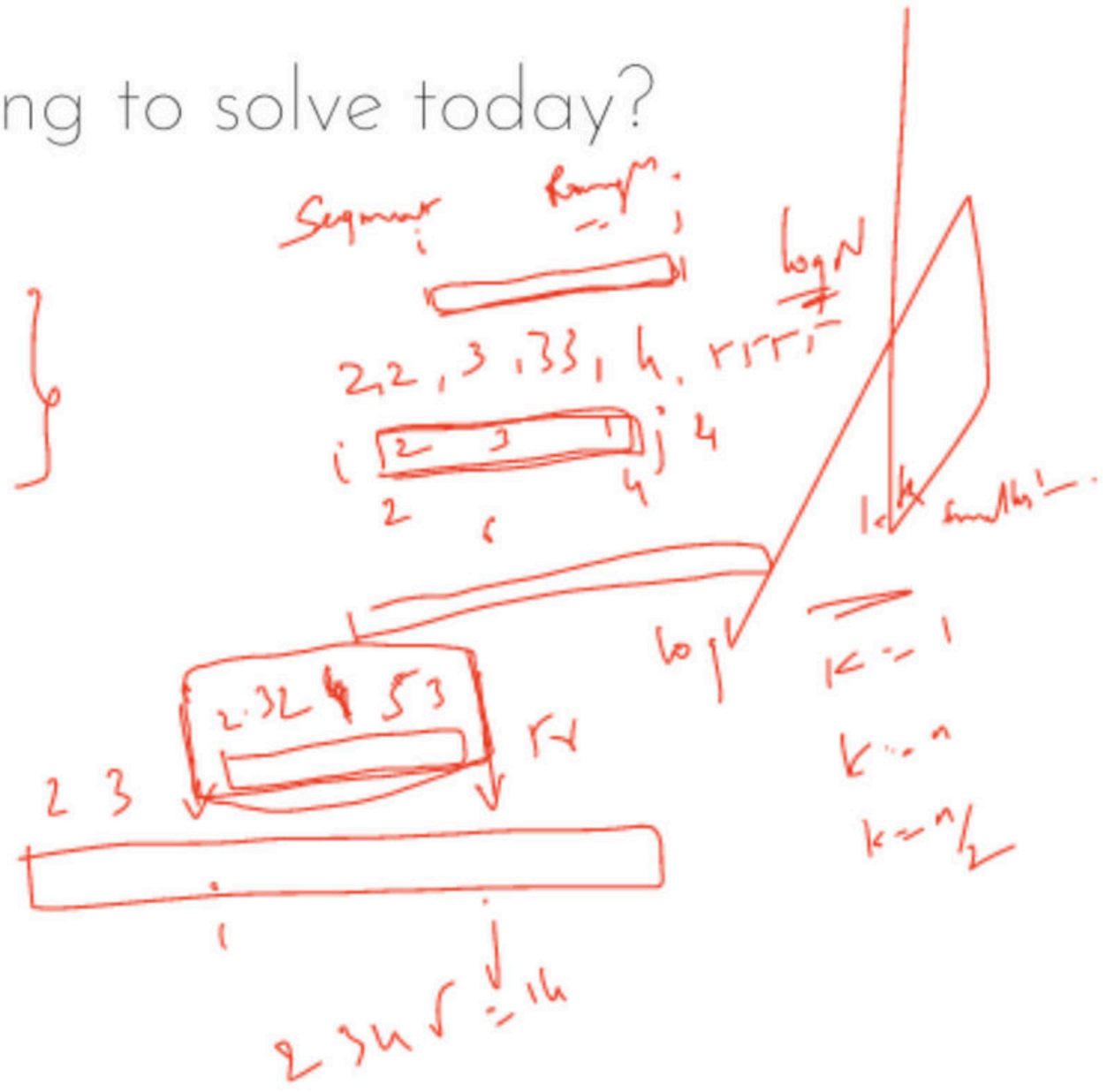

Question
What are we doing?

Objective

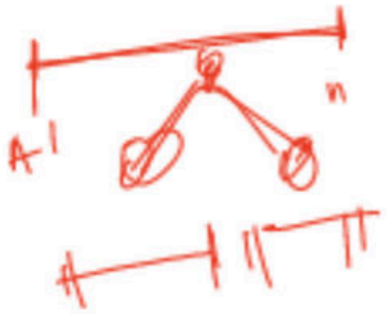
1. Class 1
   a. Fenwick Trees
2. Class 2
   a. Segment Trees
3. Class 3
   a. Persistent Segment Trees
4. **Class 4**
   a. **Competitive programming problems**

What are we going to solve today?

1. Template problems
   a. Finding min of a range
   b. Finding number of elements in a range.
   c. Finding k'th of array.
   d. Sum of unique numbers of a range.
2. Actual problems
   a. #330 Div 1 Problem - D - Codeforces
   b. SPOJ.com - Problem MKTHNUM



Finding min of a range.



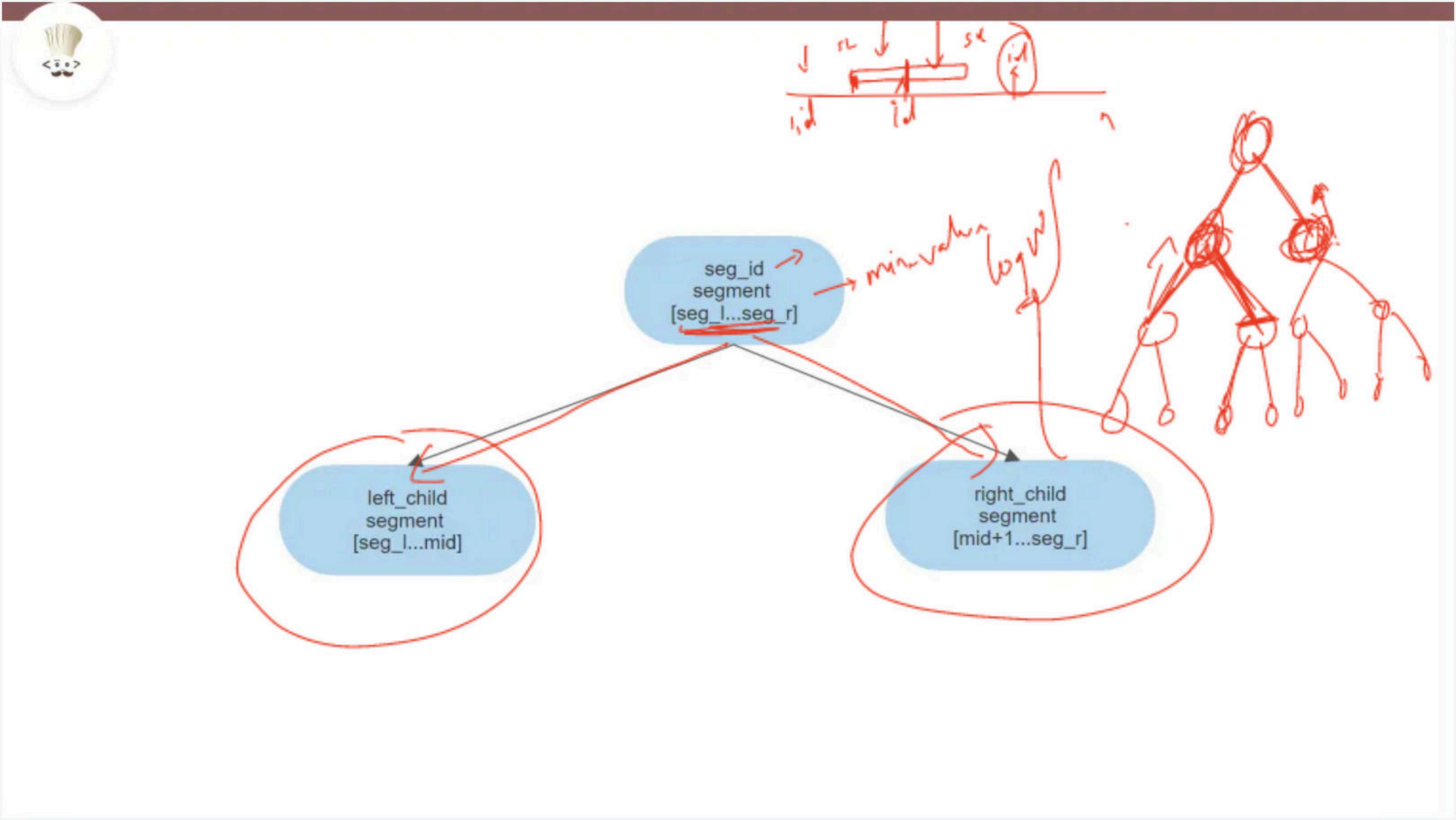

```
struct SegMin {
    int left_child_id, right_child_id;
    int min_value;
} seg[N];
int used_id_cnt = 0;

int a[N];


inline void init(int seg_id, int seg_l, int seg_r) {
    if (seg_l == seg_r) {
        seg[seg_id].min_value = a[seg_l];
        return;
    }
    seg[seg_id].left_child_id = (++used_id_cnt);
    init(seg[seg_id].left_child_id, seg_l, (seg_l + seg_r) / 2);

    seg[seg_id].right_child_id = (++used_id_cnt);
    init(seg[seg_id].right_child_id, 1 + (seg_l + seg_r) / 2, r);

    seg[seg_id].min_value = min(seg[seg[seg_id].left_child_id].min_value,
                                seg[seg[seg_id].right_child_id].min_value);
}
```

init(0, 1, n)

A) O(N)
B) O(N log N)
C) O(N²)
D) N + N/2 + N/4 + ... = 2N

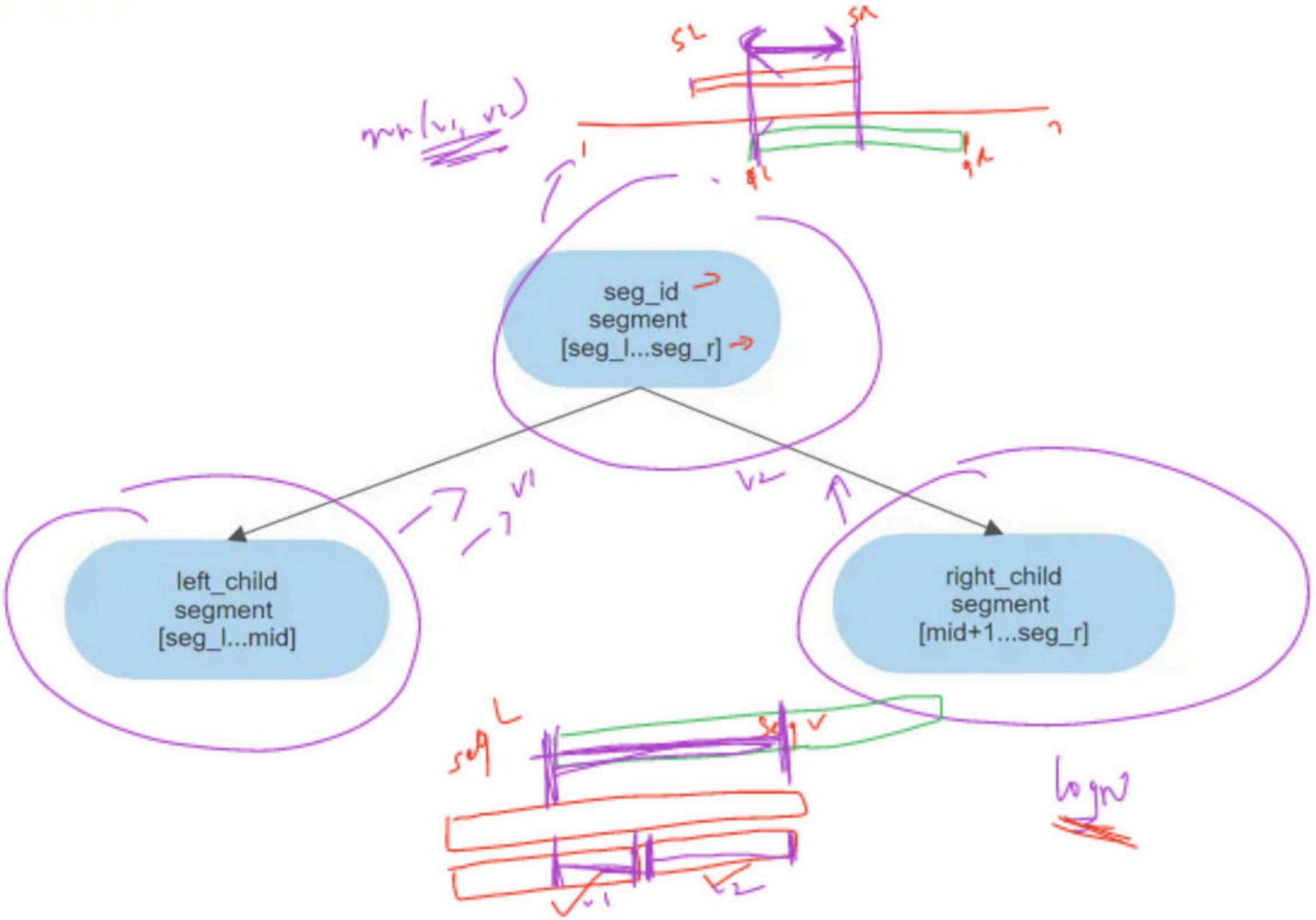
Finding min of a range.

```
inline void update(int seg_id, int seg_l, int seg_r, int id, int value) {
    if (id < seg_l || id > seg_r) return;
    if (seg_l == seg_r) {
      seg[seg_id].min_value = value;
      a[id] = value;
      return;
    }
  update(seg[seg_id].left_child_id, seg_l, (seg_l + seg_r) / 2, id, value);
  update(seg[seg_id].right_child_id, 1 + (seg_l + seg_r) / 2, seg_r, id, value);

  seg[seg_id].min_value = min(seg[seg[seg_id].left_child_id].min_value,
                              seg[seg[seg_id].right_child_id].min_value);
}
```


seg_id
segment
[seg_l...seg_r]
left_child
segment
[seg_l...mid]
right_child
segment
[mid+1...seg_r]

Finding min of a range.

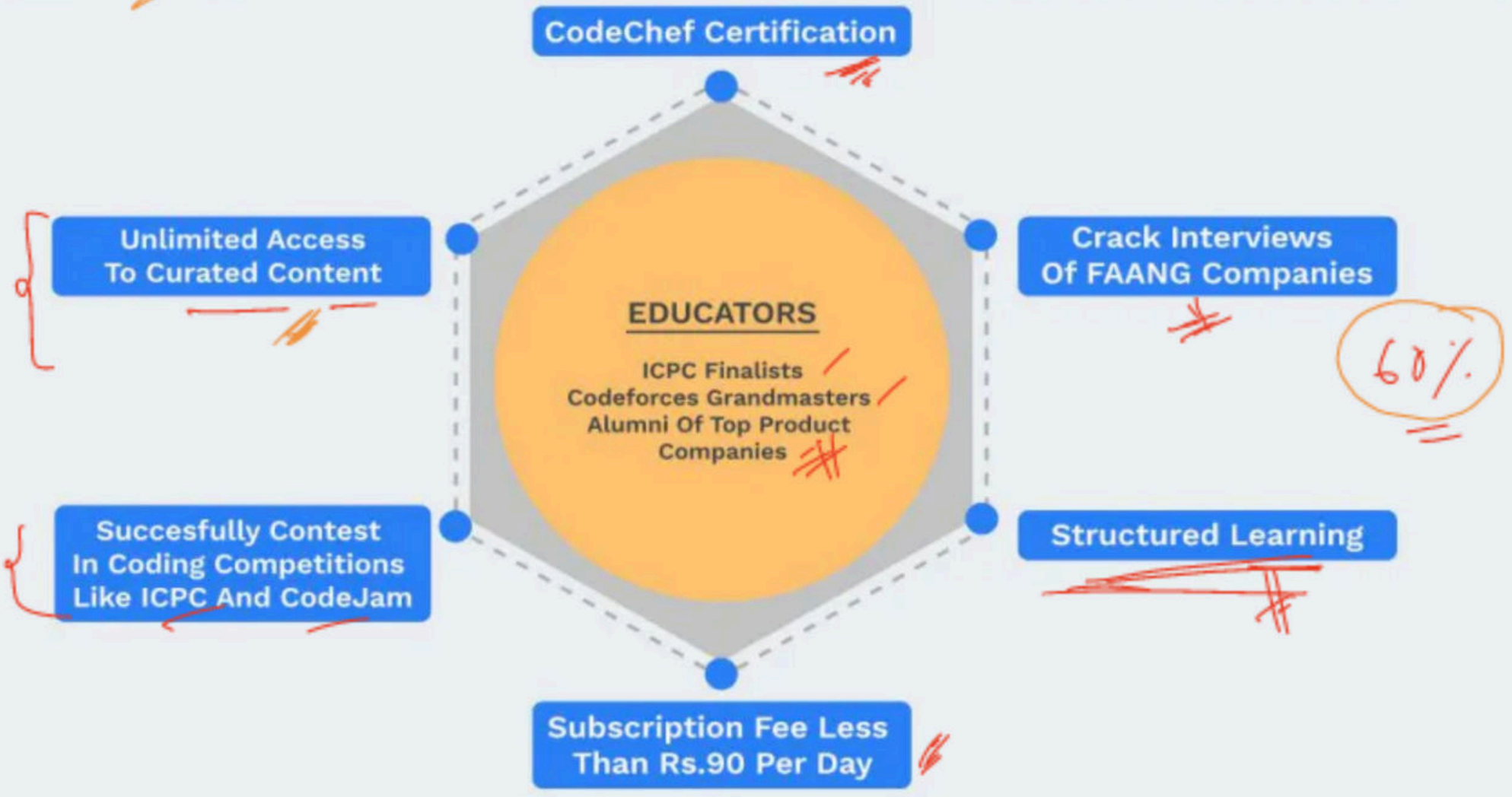
```
inline int query_min(int seg_id, int seg_l, int seg_r, int query_l, int query_r) {
    if (query_l > seg_r || query_r < seg_l) return INF;
    if (query_l <= seg_l && seg_r <= query_r) {
        return seg[seg_id].min_value;
    }
    return min(query_min(seg[seg_id].left_child_id, seg_l, (seg_l + seg_r) / 2, query_l, query_r),
               query_min(seg[seg_id].right_child_id, 1 + (seg_l + seg_r) / 2, seg_r, query_l, query_r));
}
```


seg_id
segment
[seg_l...seg_r]
left_child
segment
[seg_l...mid]
right_child
segment
[mid+1...seg_r]
logN

Questions?

Exclusive Batch Starting On 18th Jan 2021

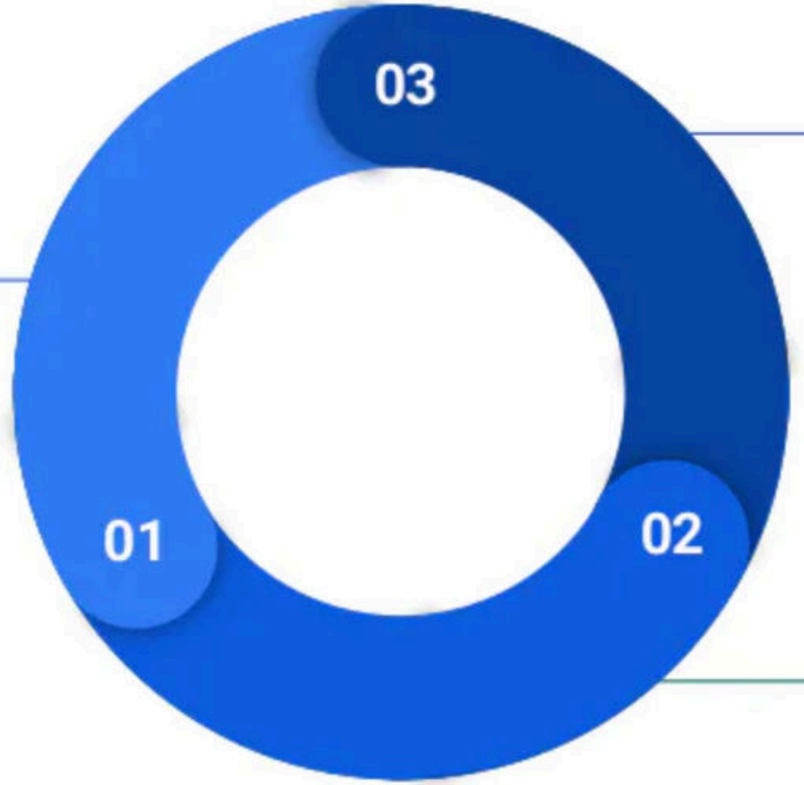
PINNACLE - Batch For Intermediates And Beginners



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

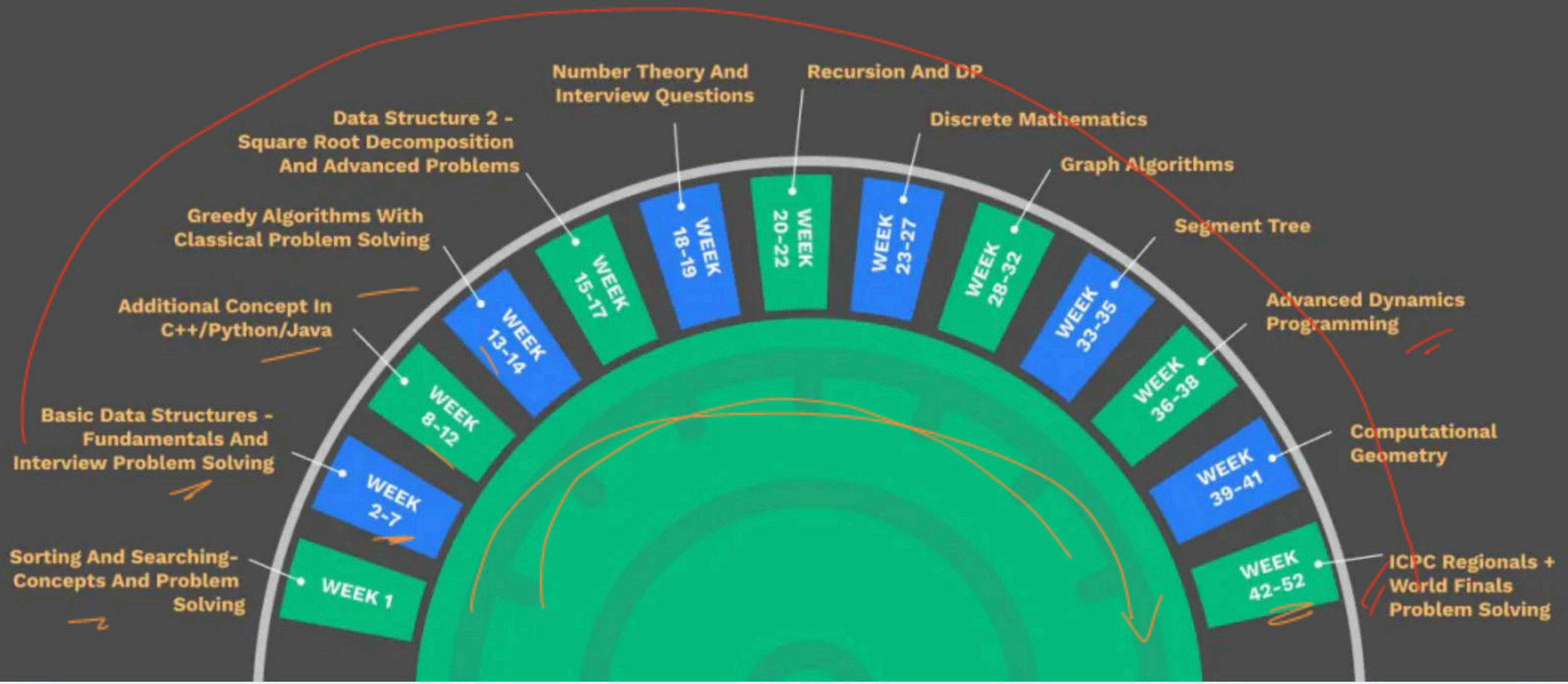
If you get stuck in any problem post class- Get your doubts resolved by our expert panel of teaching assistants and community members instantly

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 Batch Structure

PINNACLE



One Subscription and Unlimited Access to All Batches/ Courses

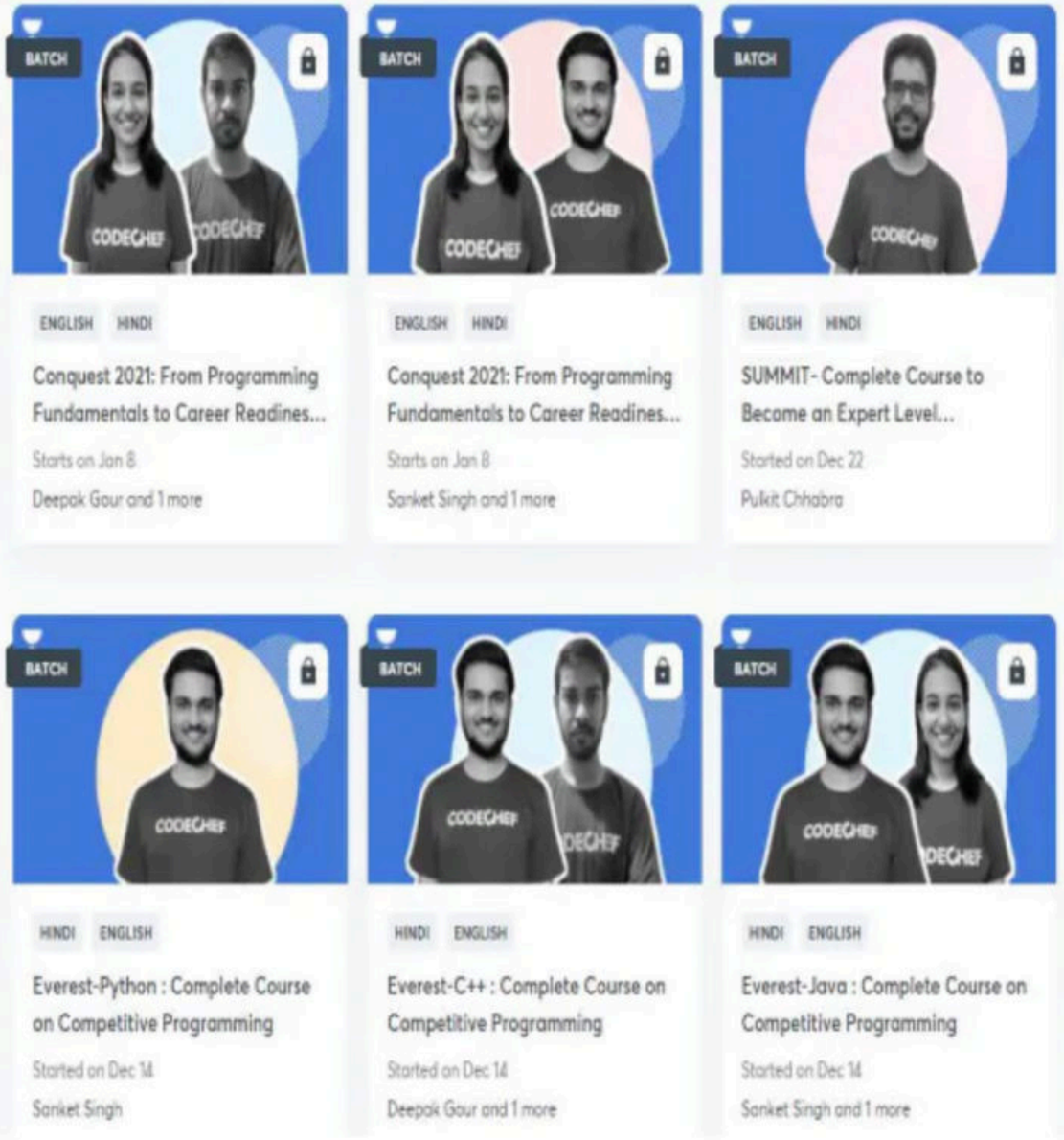
Batch Getting Live on 18th January 2021:

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.

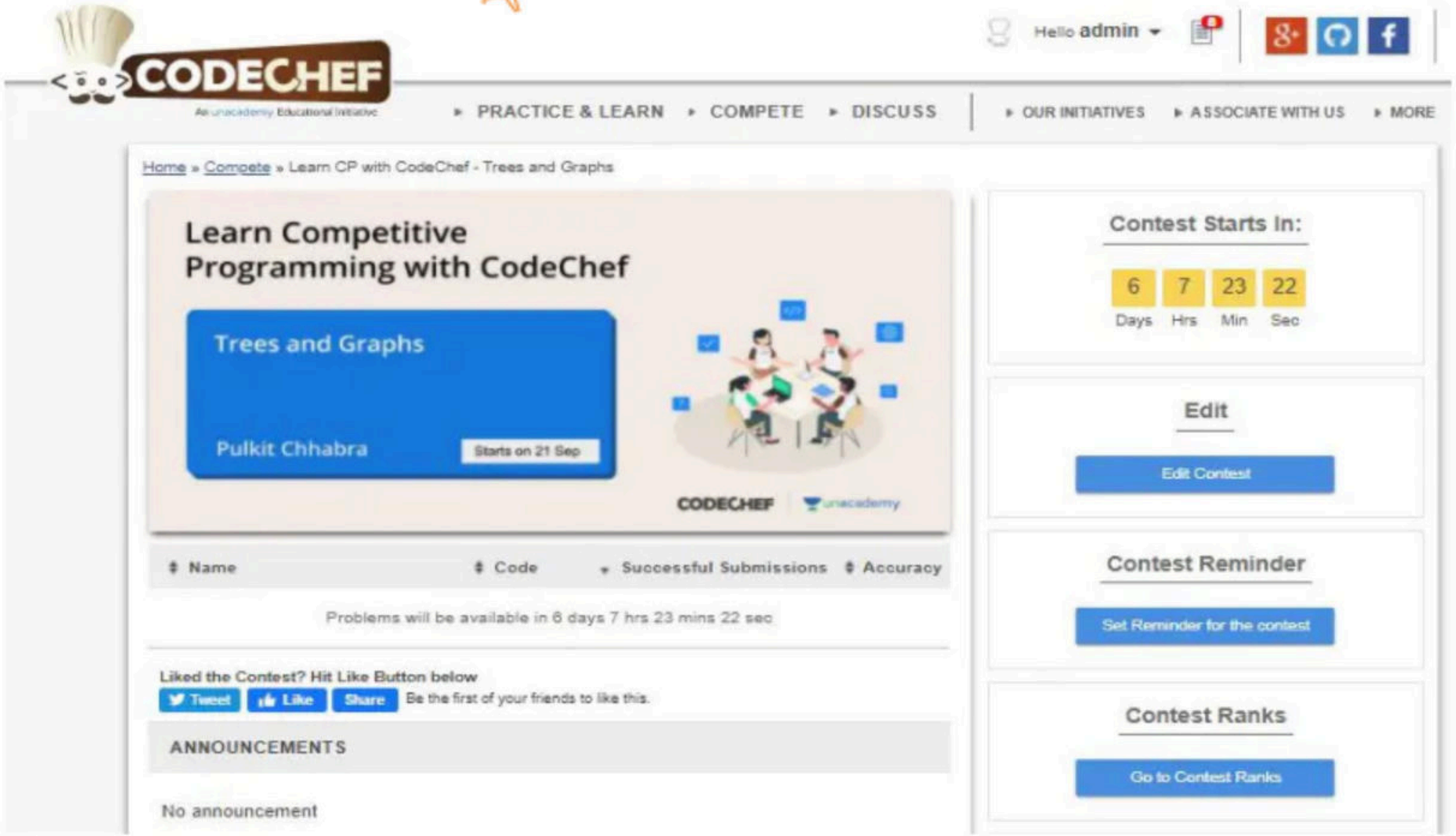
CODECHEF unacademy

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

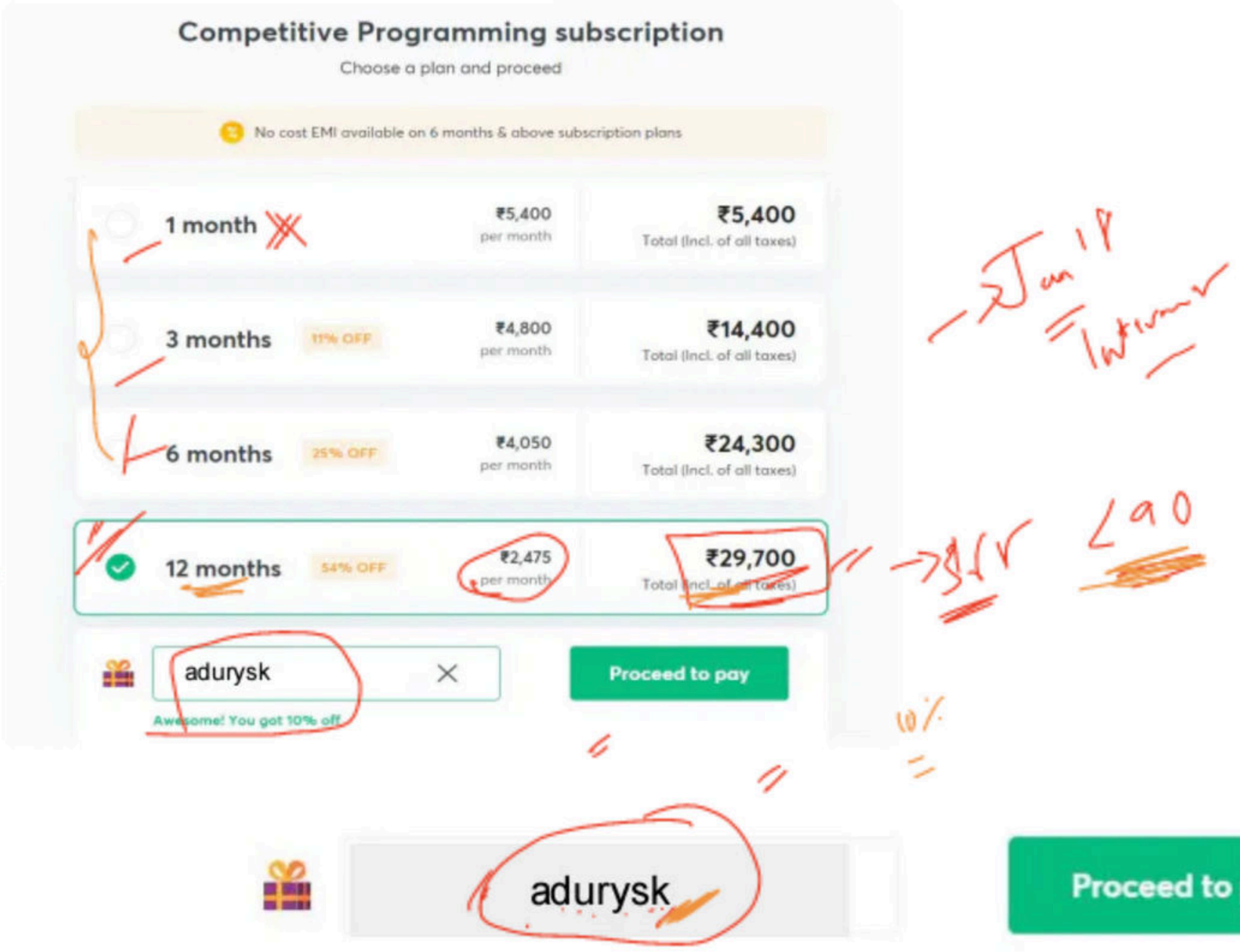
Topic	Replies	Views	Activity
About the Learn CP on Unacademy Plus category	1	6	2d

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

Course-wise Practice Problems



Flexible Subscription Plans



A) Beginner

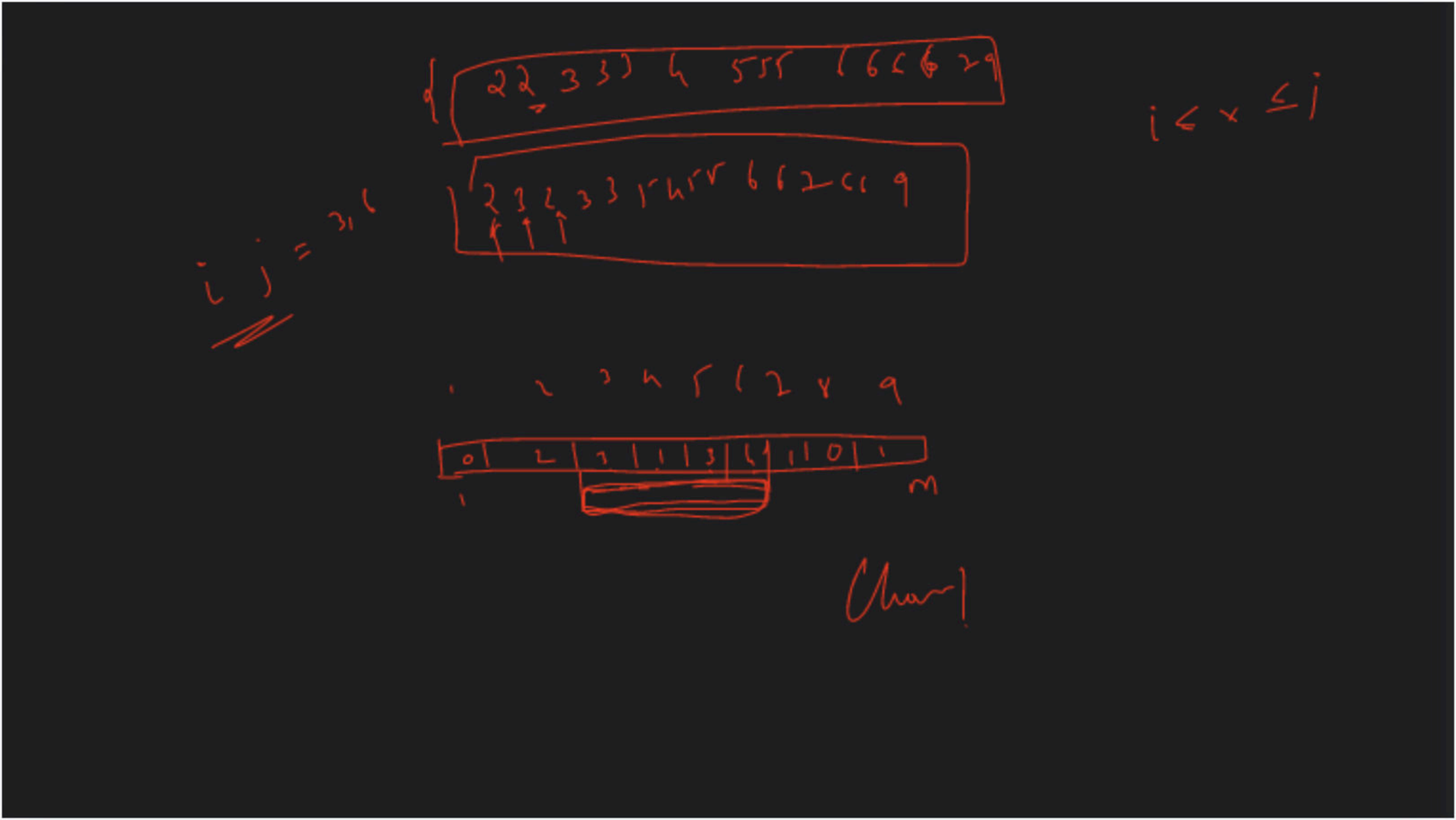
B) Never heard

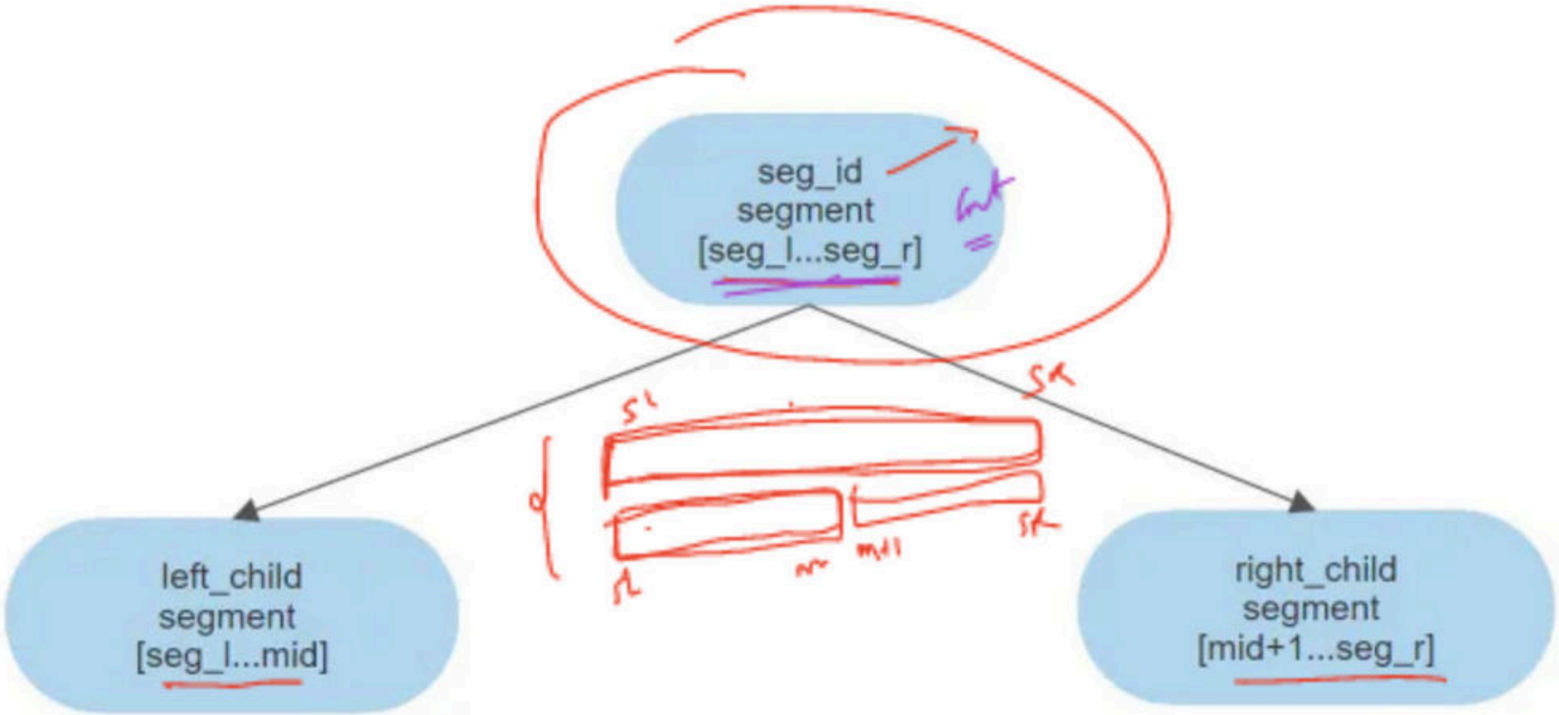
C) Intermediate

D) Advanced

Finding number of elements in a range

```
struct SegCnt {
    int left_child_id, right_child_id;
    int cnt;
} seg_cnt[N];
int used_id_cnt = 0;

int a[N];

inline void update_cnt(int seg_id, int seg_l, int seg_r, int id, int v) {
    if (id < seg_l || id > seg_r) return;
    seg_cnt[seg_id].cnt += v;
    if (seg_l != seg_r) {
        update_cnt(seg_cnt[seg_id].left_child_id, seg_l, (seg_l + seg_r) / 2, id, v);
        update_cnt(seg_cnt[seg_id].right_child_id, 1 + (seg_l + seg_r) / 2, seg_r, id, v);
    }
}
```

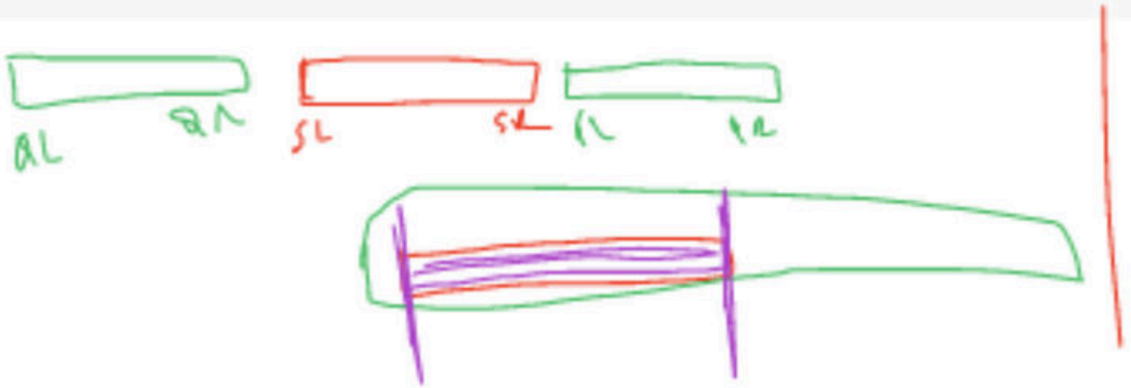


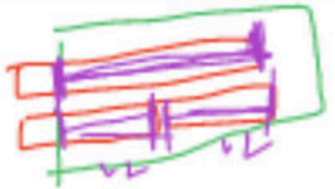

seg_id
segment
[seg_l...seg_r]
left_child
segment
[seg_l...mid]
right_child
segment
[mid+1...seg_r]

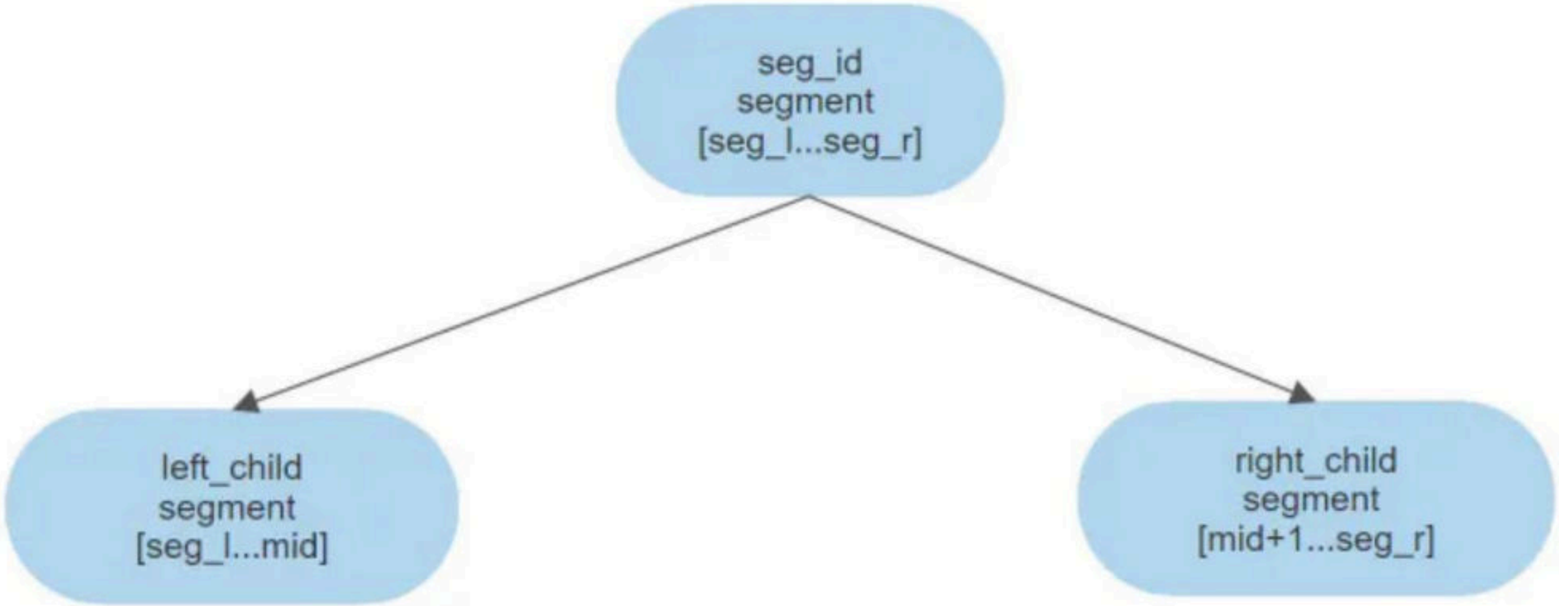
Finding number of elements in a range

query cnt (1, 1, M, i, j);

```
// Returns count of elements in the range intersection of (seg_l, seg_r) and (query_l, query_r)
inline int query_cnt(int seg_id, int seg_l, int seg_r, int query_l, int query_r) {
    if (query_l > seg_r || seg_l > query_r) return 0;
    if (query_l <= seg_l && seg_r <= query_r) return seg_cnt[seg_id].cnt;
    return query_cnt(seg_cnt[seg_id].left_child_id, seg_l, (seg_l + seg_r) / 2, query_l, query_r)
         + query_cnt(seg_cnt[seg_id].right_child_id, 1 + (seg_l + seg_r) / 2, seg_r, query_l, query_r);
}
```

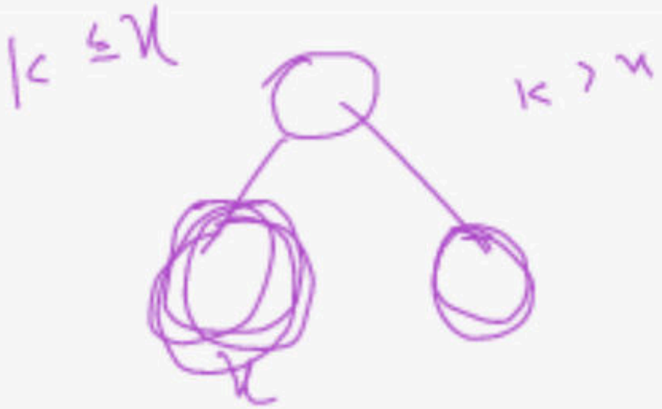


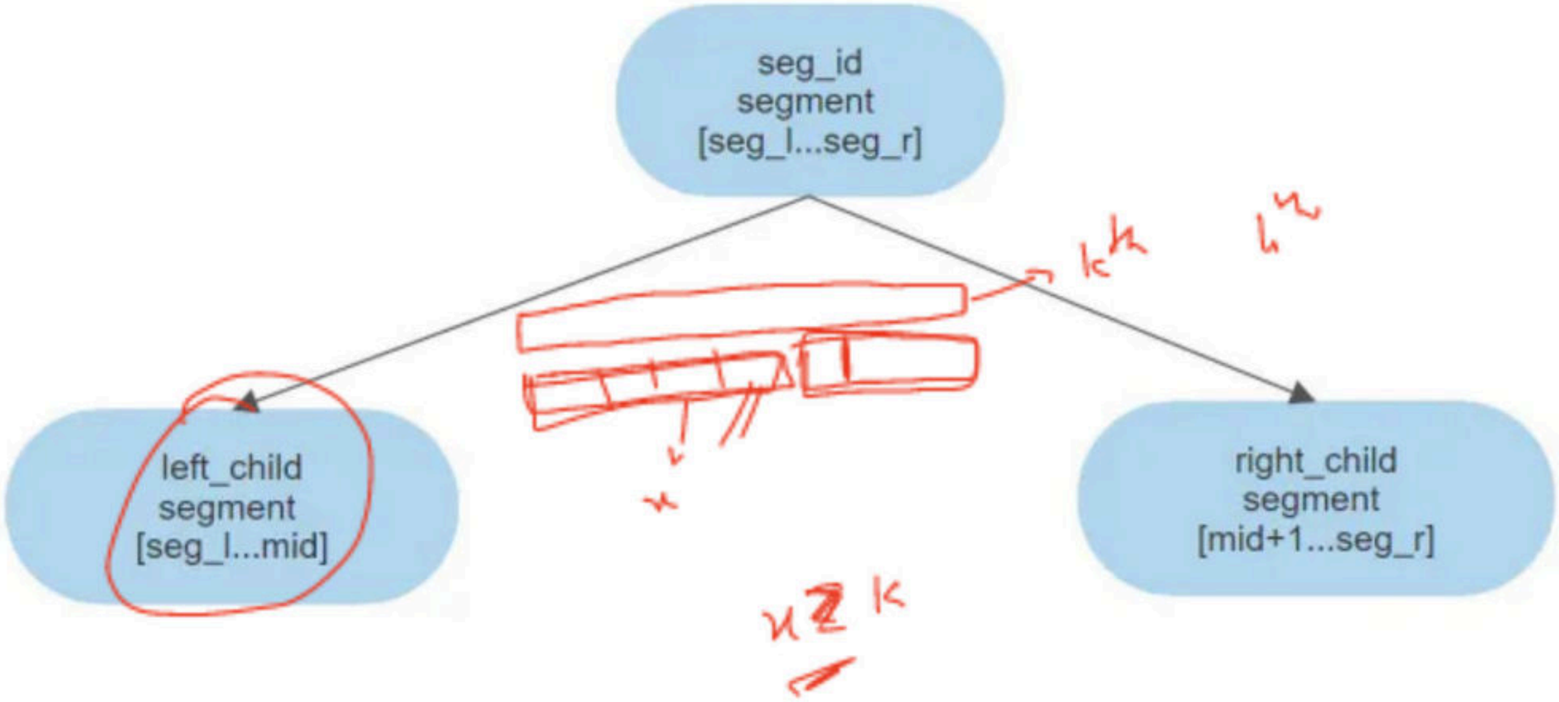



seg_id
segment
[seg_l...seg_r]
left_child
segment
[seg_l...mid]
right_child
segment
[mid+1...seg_r]

Finding k'th element in a range

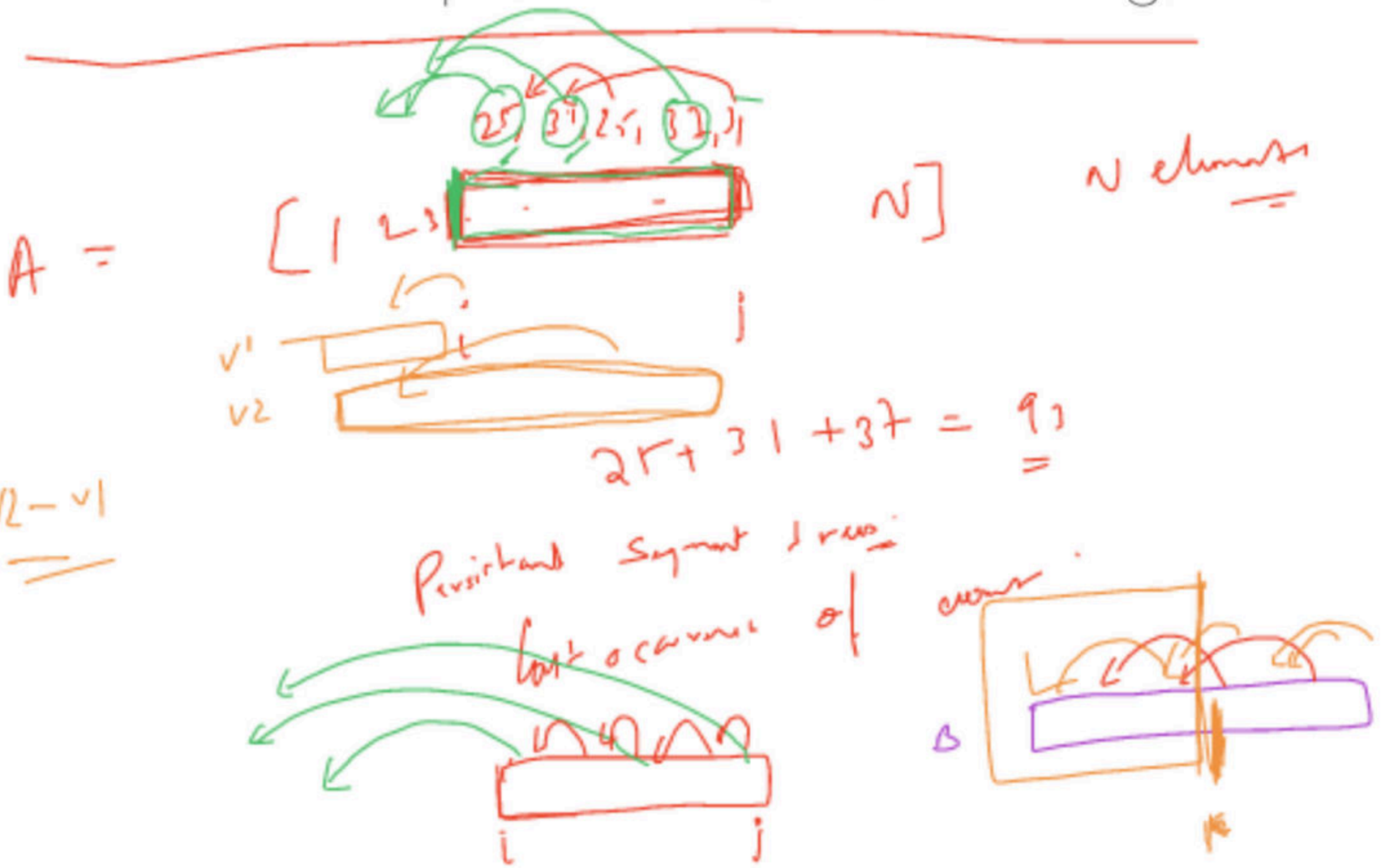
```
// Returns k'th element in the range (seg_l, seg_r)
inline int query_kth(int seg_id, int seg_l, int seg_r, int k) {
    if (seg_r == seg_l) {
        return seg_l;
    }
    int left_cnt = seg_cnt[seg_cnt[seg_id].left_child_id].cnt;
    if (left_cnt >= k) {
        return query_kth(seg_cnt[seg_id].left_child_id, seg_l, (seg_l + seg_r) / 2, k);
    } else {
        return query_kth(seg_cnt[seg_id].right_child_id, 1 + (seg_l + seg_r) / 2, seg_r, k - left_cnt);
    }
}
```




seg_id
segment
[seg_l...seg_r]
left_child
segment
[seg_l...mid]
right_child
segment
[mid+1...seg_r]

Questions?

Sum of unique numbers of a range



Sum of unique numbers of a range

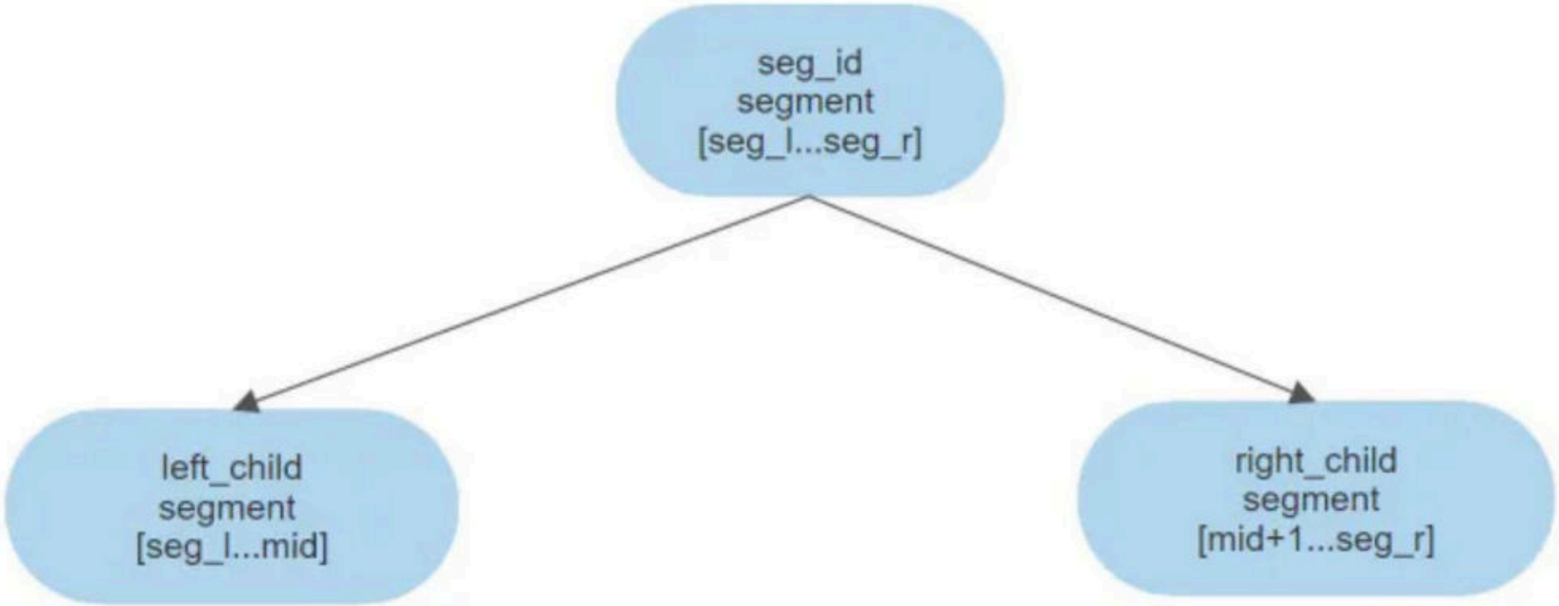
```
int last_occ[N];
int a[N];

inline void solve () {
    int n;
    cin >> n;
    rep (i, 1, n) {
        cin >> a[i];
    }
    root_id[0] = init(0, n);
    for (int i = 1; i <= n; ++i) {
        root_id[i] = update(root_id[i - 1], 0, n, last_occ[a[i]], a[i]);
        last_occ[a[i]] = i;
    }
    int q;
    cin >> q;
    while (q--) {
        int l, r;
        cin >> l >> r;
        int prefix_sum_1 = query(root_id[r], 0, n, l - 1);
        int prefix_sum_2 = query(root_id[l - 1], 0, n, l - 1);
        cout << prefix_sum_1 - prefix_sum_2 << endl;
    }
}
```

Sum of unique numbers of a range

```
Struct SegSum {
    int left_child_id;
    int right_child_id;
    int sum;
} seg_sum[S];

int root_id[N];
int left_child_id[N];
int right_child_id[N];
int used_seg_cnt = 0;

inline int init(int l, int r) {
    int seg_id = ++used_seg_cnt;
    if (l < r) {
        seg_sum[seg_id].left_child_id = init(l, (l + r) / 2);
        seg_sum[seg_id].right_child_id = init(1 + (l + r) / 2, r);
    }
    return seg_id;
}
```

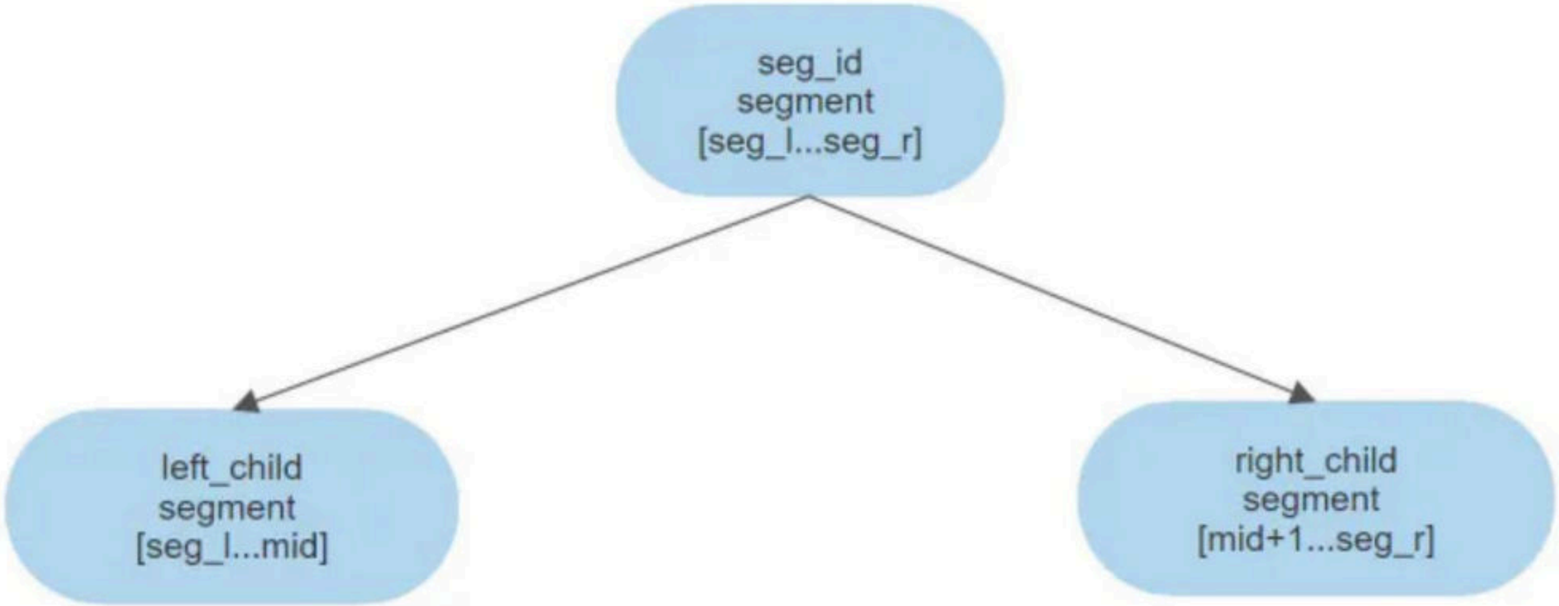
Sum of unique numbers of a range

```
inline int make_parent(int left_child_id, int right_child_id) {
    int new_seg_id = ++used_seg_cnt;
    seg_sum[new_seg_id].sum = seg_sum[left_child_id].sum + seg_sum[right_child_id].sum;
    seg_sum[new_seg_id].left_child_id = left_child_id;
    seg_sum[new_seg_id].right_child_id = right_child_id;
    return new_seg_id;
}

inline int make_leaf(int v) {
    int new_seg_id = ++used_seg_cnt;
    seg_sum[new_seg_id].sum = v;
    return new_seg_id;
}

inline int update(int seg_id, int seg_l, int seg_r, int id, int v) {
    if (id < seg_l || id > seg_r) return seg_id;

    if (seg_l == seg_r) return make_leaf(id, v);

    return make_parent(update(seg_sum[seg_id].left_child_id, seg_l, (seg_l + seg_r) / 2, id, v),
                       update(seg_sum[seg_id].right_child_id, 1 + (seg_l + seg_r) / 2, seg_r, id, v));
}
```


seg_id
segment
[seg_l...seg_r]
left_child
segment
[seg_l...mid]
right_child
segment
[mid+1...seg_r]

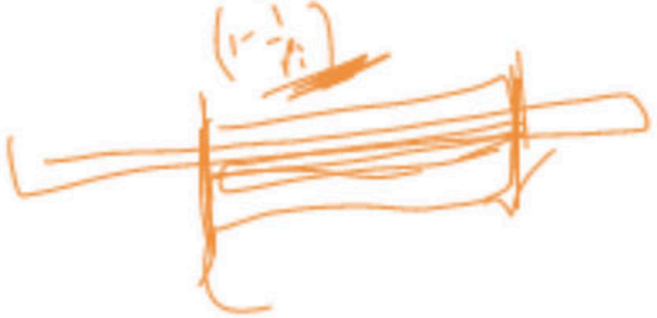
Sum of unique numbers of a range

```
int query(int seg_id, int seg_l, int seg_r, int query_prefix_id) {
    if (query_prefix_id < seg_l) return 0;
    if (query_prefix_id >= seg_r) return seg_sum[seg_id].sum;
    return query(seg[seg_id].left_child_id, seg_l, (seg_l + seg_r) / 2, query_prefix_id)
         + query(seg[seg_id].right_child_id,  1 + (seg_l + seg_r) / 2, seg_r, query_prefix_id);
}
```


seg_id
segment
[seg_l...seg_r]
left_child
segment
[seg_l...mid]
right_child
segment
[mid+1...seg_r]

Questions?

#330 Div 1 Problem - D - Codeforces



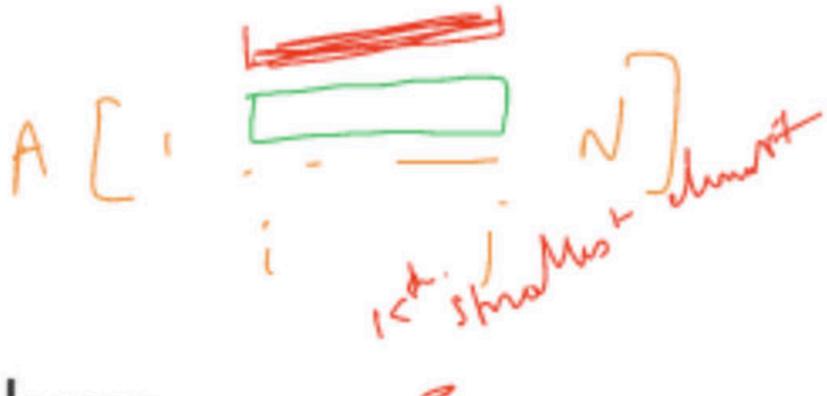
D. REQ

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Today on a math lesson the teacher told Vovochka that the Euler function of a positive integer $\varphi(n)$ is an arithmetic function that counts the positive integers less than or equal to n that are relatively prime to n. The number 1 is coprime to all the positive integers and $\varphi(1) = 1$.

Now the teacher gave Vovochka an array of n positive integers $a_1, a_2, ..., a_n$ and a task to process q queries $l_i\ r_i$ — to calculate and print $\varphi\left(\prod_{i=l}^{r} a_i\right)$ modulo $10^9 + 7$. As it is too hard for a second grade school student, you've decided to help Vovochka.

Questions?



MKTHNUM - K-th Number

#sorting #tree

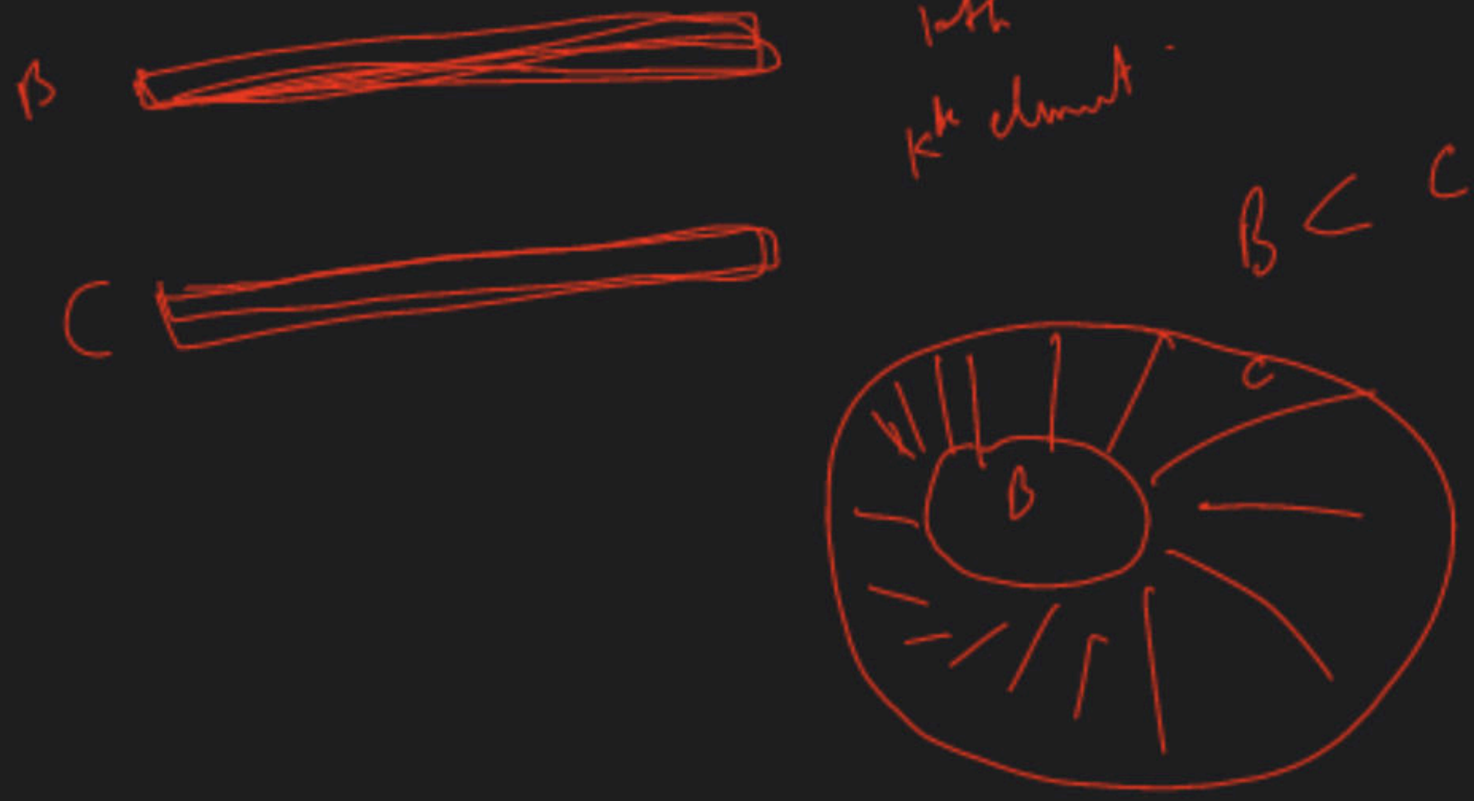
English

Vietnamese

You are working for Macrohard company in data structures department. After failing your previous task about key insertion you were asked to write a new data structure that would be able to return quickly k-th order statistics in the array segment.

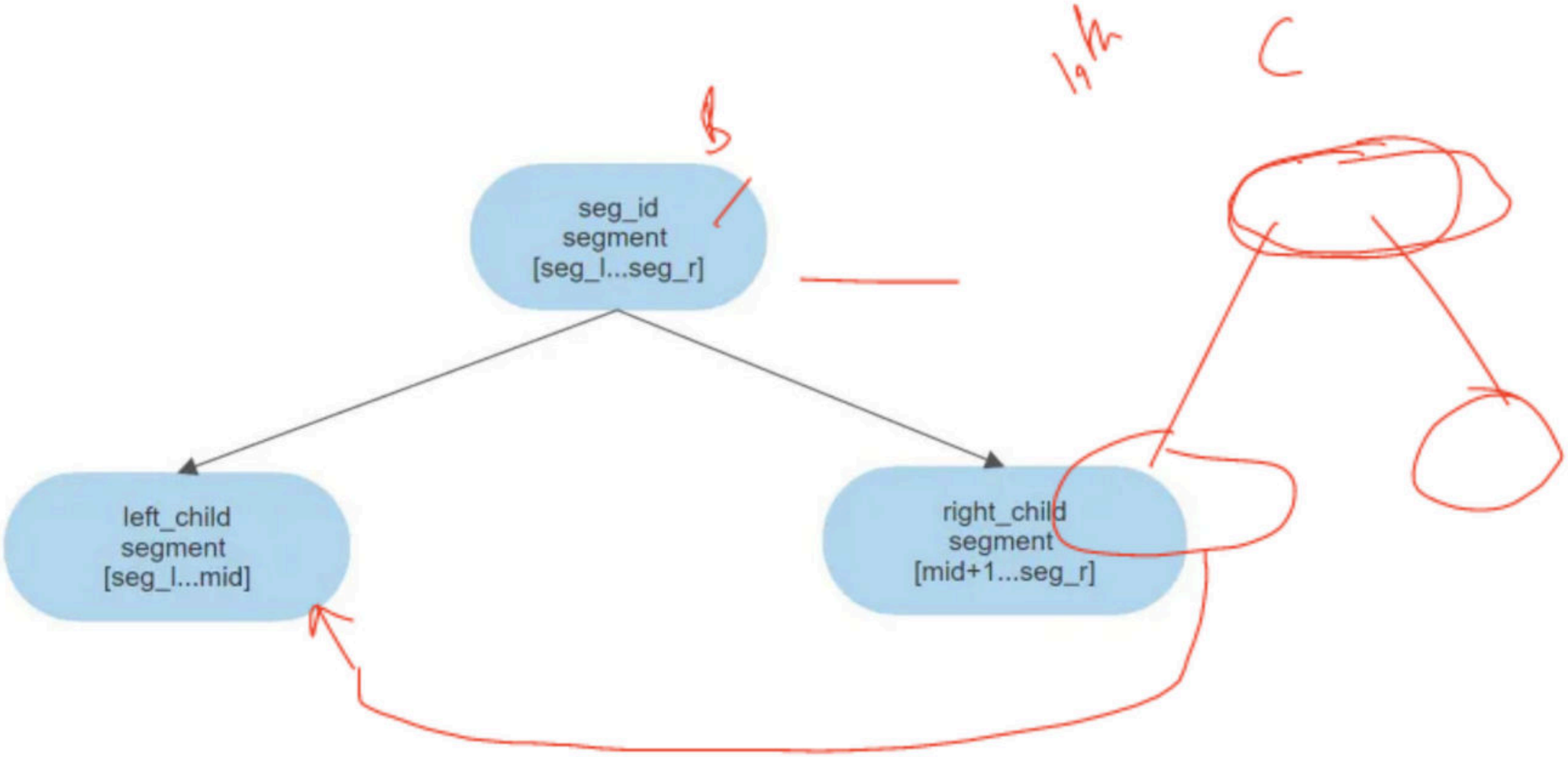
That is, given an array a[1 ... n] of different integer numbers, your program must answer a series of questions Q(i, j, k) in the form: "What would be the k-th number in a[i ... j] segment, if this segment was sorted?"

For example, consider the array a = (1, 5, 2, 6, 3, 7, 4). Let the question be Q(2, 5, 3). The segment a[2 ... 5] is (5, 2, 6, 3). If we sort this segment, we get (2, 3, 5, 6), the third number is 5, and therefore the answer to the question is 5.


B
C
B < C
B
C

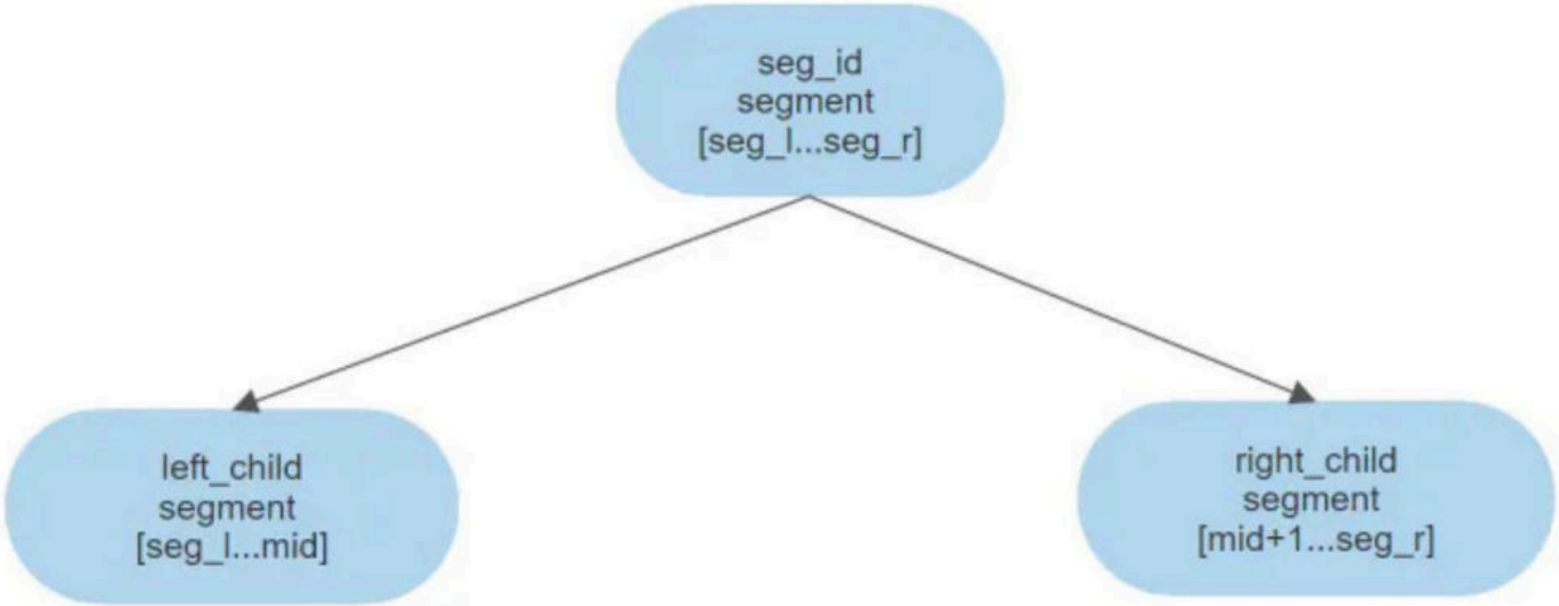
MKTHNUM

```
inline int query(int seg_id1, int seg_id2, int seg_l, int seg_r, int k) {
    if (seg_l == seg_r) return seg_l;
    int left_cnt = seg_cnt[seg_cnt[seg_id1].left_child_id].cnt
                 - seg_cnt[seg_cnt[seg_id2].left_child_id].cnt;
    if (left_cnt >= k) {
        return query_kth(seg_cnt[seg_id1].left_child_id, seg_cnt[seg_id2].left_child_id, seg_l, (seg_l + se
g_r) / 2, k);
    } else {
        return query_kth(seg_cnt[seg_id1].right_child_id, seg_cnt[seg_id2].right_child_id, 1 + (seg_l + seg
_r) / 2, seg_r, k - left_cnt);
    }
}
```


seg_id
segment
[seg_l...seg_r]
left_child
segment
[seg_l...mid]
right_child
segment
[mid+1...seg_r]

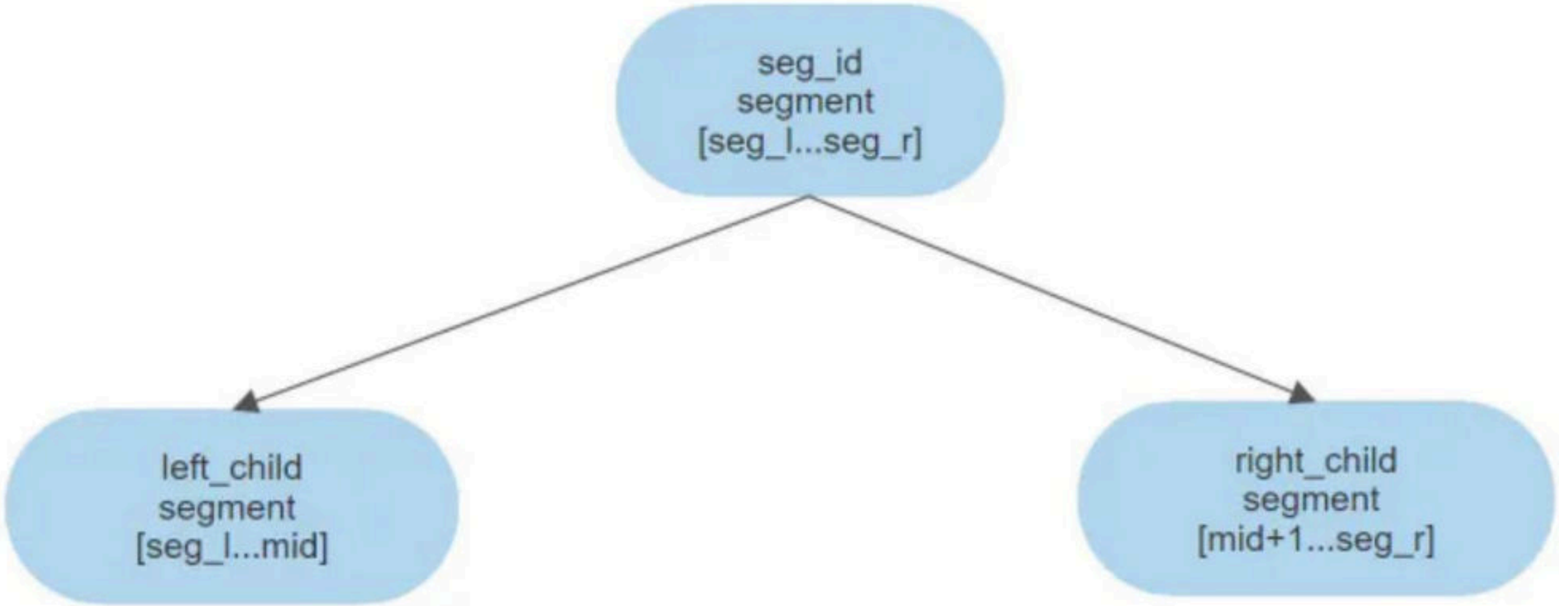
MKTHNUM

```
inline int make_parent(int left_child_id, int right_child_id) {
    int new_seg_id = ++used_seg_cnt;
    seg_sum[new_seg_id].sum = seg_sum[left_child_id].sum + seg_sum[right_child_id].sum;
    seg_sum[new_seg_id].left_child_id = left_child_id;
    seg_sum[new_seg_id].right_child_id = right_child_id;
    return new_seg_id;
}

inline int make_leaf(int v) {
    int new_seg_id = ++used_seg_cnt;
    seg_sum[new_seg_id].sum = v;
    return new_seg_id;
}

inline void update(int seg_id, int seg_l, seg_r, int id) {
    if (id < seg_l || id > seg_r) return seg_id;

    if (seg_l == seg_r) return make_leaf(id, seg_cnt[seg_id].cnt + 1);

    return make_parent(update(seg_sum[seg_id].left_child_id, seg_l, (seg_l + seg_r) / 2, id),
                       update(seg_sum[seg_id].right_child_id, 1 + (seg_l + seg_r) / 2, seg_r, id));
}
```


seg_id
segment
[seg_l...seg_r]
left_child
segment
[seg_l...mid]
right_child
segment
[mid+1...seg_r]

MKTHNUM

```
inline void solve() {
    int n;
    cin >> n;
    root_id[0] = init(1, M);
    for (int i = 1; i <= n; ++i) {
        cin >> a[i];
        cnt[a[i]]++;
        root_id[i] = update(root_id[i - 1], 1, M, a[i]);
    }
    int q;
    cin >> q;
    while (q--) {
        int i, j, k;
        cin >> i >> j >> k;
        cout << query(root_id[j], root_id[i - 1], 1, M, k) << endl;
    }
}
```


seg_id
segment
[seg_l...seg_r]
left_child
segment
[seg_l...mid]
right_child
segment
[mid+1...seg_r]

Questions?

adury.suvg@gmail.com