

# Meet In The Middle - I

Special class



# Vivek Chauhan

< Educator Image >

- < Educator Credentails >
-

2021 : The Year To QUIT PROCRASTINATING And LEARN CODING

Join Our Exclusive **BATCHES**



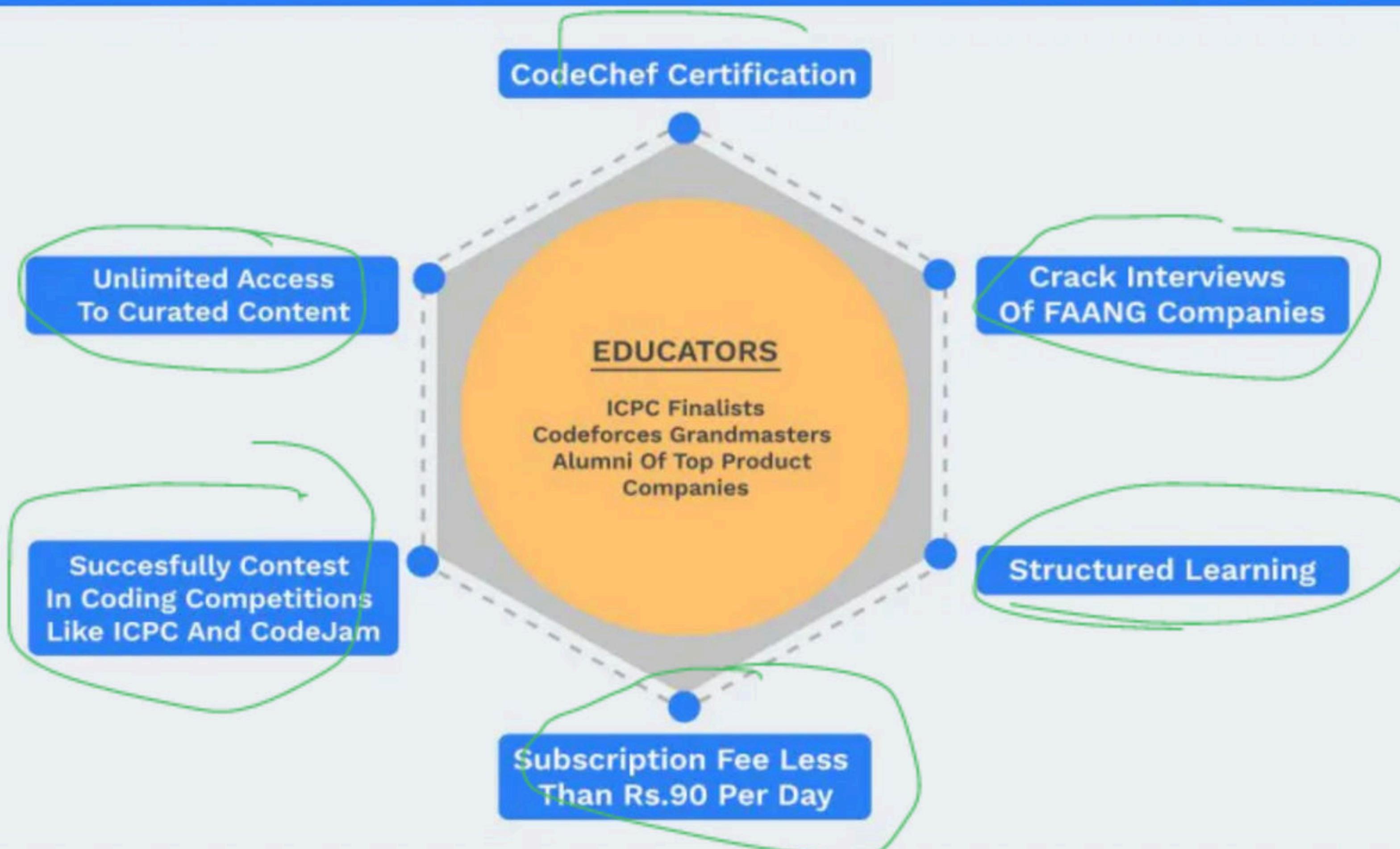
**PINNACLE: COMPREHENSIVE AND CONCISE TRACK TO BECOME AN EXPERT GOING LIVE ON 18TH JAN 2021**

- 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

# Exclusive Batch Starting On 18th Jan 2021

## PINNACLE - Batch For Intermediates And Beginners

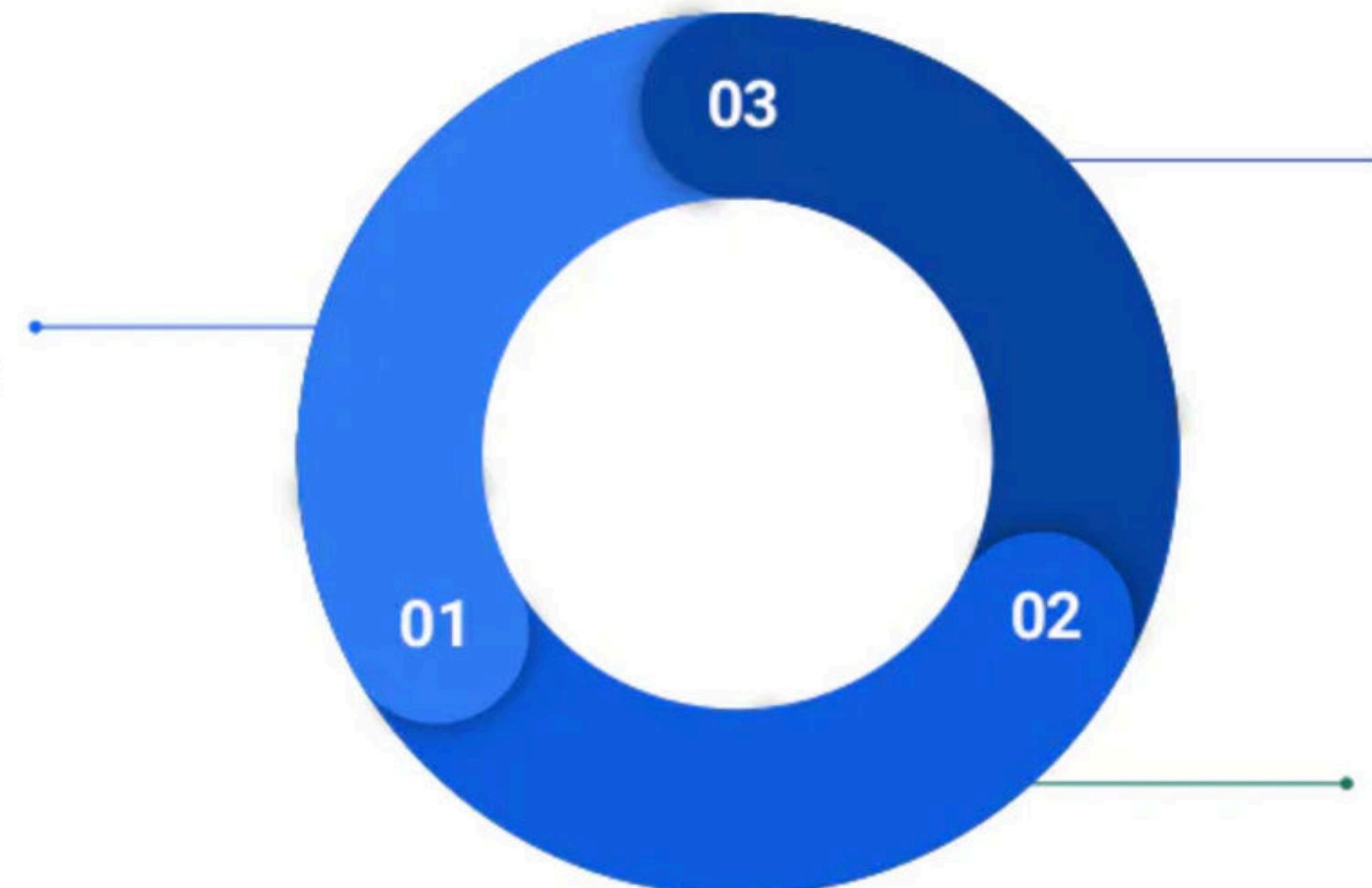




# What you will get

## Live Interactive Classes

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



## Doubt Support

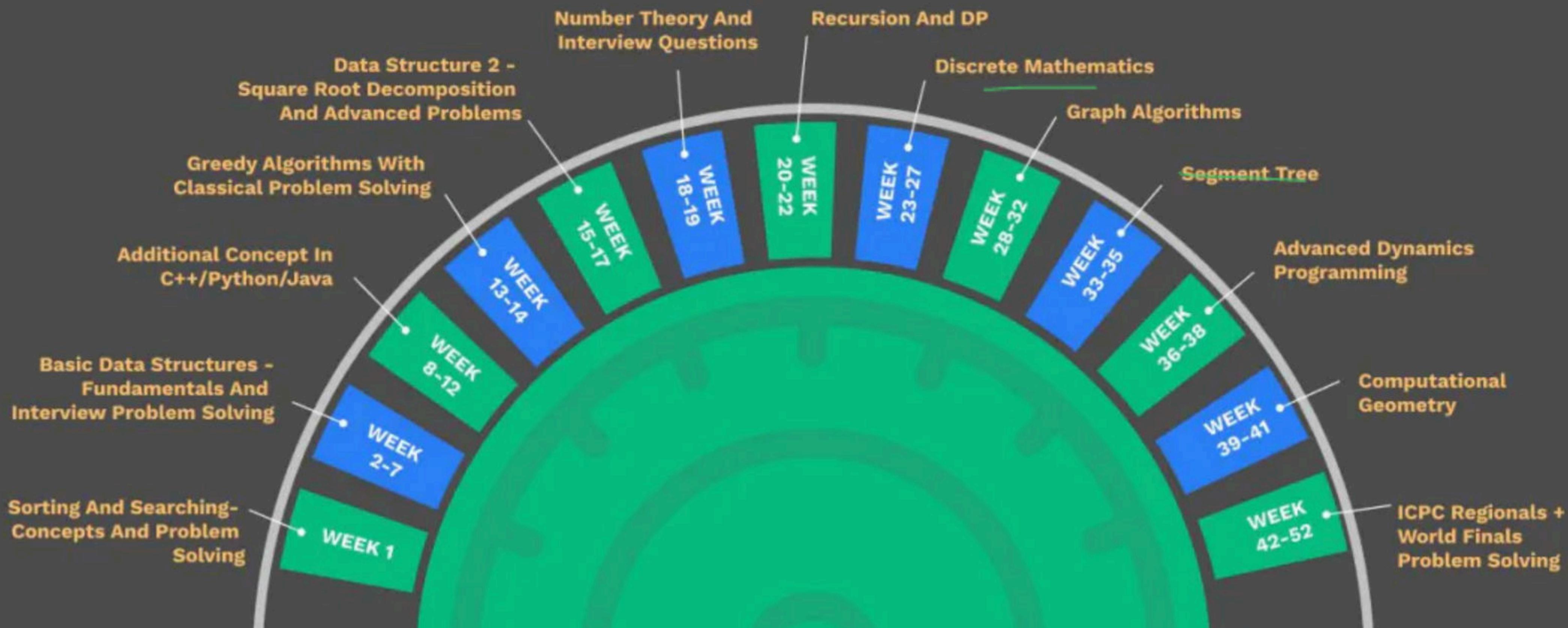
If you get stuck in any problem post class-  
Get your doubts resolved by our expert panel of teaching assistants and community members instantly

## 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**



ENGLISH HINDI

Conquest 2021: From Programming Fundamentals to Career Readiness...  
Starts on Jan 8  
Deepak Gour and 1 more

ENGLISH HINDI

Conquest 2021: From Programming Fundamentals to Career Readiness...  
Starts on Jan 8  
Sanket Singh and 1 more

ENGLISH HINDI

SUMMIT- Complete Course to Become an Expert Level...  
Started on Dec 22  
Pulkit Chhabra



HINDI ENGLISH

Everest-Python : Complete Course on Competitive Programming  
Started on Dec 14  
Sanket Singh

HINDI ENGLISH

Everest-C++ : Complete Course on Competitive Programming  
Started on Dec 14  
Deepak Gour and 1 more

HINDI ENGLISH

Everest-Java : Complete Course on Competitive Programming  
Started on Dec 14  
Sanket Singh and 1 more



# 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

**Murugappan S**

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

**Nishchay Manwani**

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

**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



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

Hello admin

CODECHEF

An Unacademy Educational Initiative

PRACTICE & LEARN COMPETE DISCUSS

OUR INITIATIVES ASSOCIATE WITH US MORE

Home » Compete » Learn CP with CodeChef - Trees and Graphs

## Learn Competitive Programming with CodeChef

### Trees and Graphs

Pulkit Chhabra Starts on 21 Sep

CODECHEF unacademy

# Name	# Code	* Successful Submissions	* Accuracy
--------	--------	--------------------------	------------

Problems will be available in 6 days 7 hrs 23 mins 22 sec

Liked the Contest? Hit Like Button below

[Tweet](#) [Like](#) [Share](#) Be the first of your friends to like this.

#### ANNOUNCEMENTS

No announcement

Contest Starts In:

6 Days 7 Hrs 23 Min 22 Sec

Edit

Edit Contest

Contest Reminder

Set Reminder for the contest

Contest Ranks

Go to Contest Ranks



# 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)



vivek\_1998299



Proceed to pay

Awesome! You got 10% off!



vivek\_1998299

Proceed to pay





Meet in the Middle

Vivek Chauhan  
(vivek\_1998299)



# Prerequisites

None

You are given an array **A** of size **N**.

Find out if there are any four numbers in the array that sum up to zero (the same element can be used multiple times)

**Constraints:**

$1 \leq N \leq 1000$

$-10^9 \leq A_i \leq 10^9$

$$f(a, b, c) = A^{\{a\} \cup \{b\} \cup \{c\}}$$

$$6()$$

$$2 > 3$$

$$6(3 \times n^3)$$

$$= 3 \times 10^9$$

N = 4

A = [1, 3, -1, -2]

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & -1 & -2 \end{bmatrix}$$

$$1 + 3 + -1 + -2 \Rightarrow 1 \neq 0$$

Yes

$$1 + 3 + -2 + -2 \Rightarrow 0 \neq 0$$

$$1 + -1 + 1 + -1 = 0$$

$$N = \underline{5}$$

$$A = [1, 3, 2, 6, 0]$$

$$A = \underline{\underline{1 \ 2 \ 3 \ 4 \ 5}} \\ \underline{\underline{1 \ 3 \ 2 \ 6 \ 0}}$$

$$1, 3, 2, 6 > 0$$

$$1, 3, 2, 0 > 0$$

Yes

$$b, v, w, 0 = D$$

N = 3

A = [1, 3, -10]

N √

$$(3+3+3) + \underline{-10}$$

$$\Rightarrow -1 \pm \text{D}$$

N = 6

A = [2, 3, 1, -4, -1, 30]

A = 1 2 3 4 5 6  
2 [-3] 1 [-4] -1 30

What will be the answer?

2, 3, -4, -1 = 0

1, -1, 1, 1 => 0

$N = 6$        $O(N^4) = (O^3)^4 = O^{12}$  operations  
 $A = [2, 3, 1, -4, -1, 30]$  such that  
 $\begin{pmatrix} a & b & c & d \\ 1 & 1 & 1 & 1 \end{pmatrix}$

$O(1)$  for ( $a=1$ ;  $a \leq b$ ;  $a++$ ) {  
 10<sup>12</sup>  $O(N)$  for ( $b=1$ ;  $b \leq N$ ;  $b++$ ) {  
 $\frac{10^8}{10^8} = 10^4$   $O(N)$  for ( $c=1$ ;  $c \leq N$ ;  $c++$ ) {  
 $\delta(10^8) = O(N)$  for ( $d=1$ ;  $d \leq N$ ;  $d++$ ) {  
 10<sup>12</sup> if ( $A[a] + A[b] + A[c] + A[d] = 0$ )  
 $\Rightarrow \delta(10^8) = O(N)$   
 return 1  
 return 0;

$N = 6$

$A = [2, 3, 1, -4, -1, 30]$

Main problem

$$A[a] + A[b] + A[c] + A[d] = 0$$

$$(A[a] + A[b]) + (A[c] + A[d]) = 0$$

$$\left\{ \begin{array}{l} (a, b, c, d) \text{ such that} \\ A[a] + A[b] + A[c] + A[d] = 0 \end{array} \right.$$

$$A[a] +$$
  
$$\sum_{i=1}^4$$

Find all possible  
sums for tuple  
 $(a, b)$

Find all possible  
sums for tuple  
 $(c, d)$

$$(N[a] + A[b]) \cdot (N[c] + A[d]) = 0$$

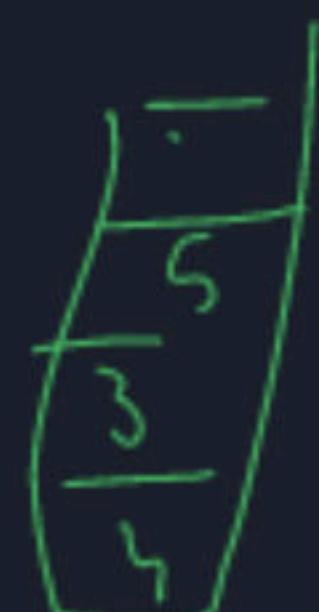
↓  
→ (c, d)

$$N = [6, 2, 3, 4, 5, 6]$$

$$A = [2, 3, 1, -4, -1, 30]$$

$$(a, b) \quad (\underline{5}, \underline{1}) \Rightarrow 2$$

$$\begin{aligned} (1, 1) &\Rightarrow 4 \neq 5 \\ (1, 3) &\Rightarrow 3 \neq 5 \\ (1, 2) &\Rightarrow 5 \checkmark \end{aligned}$$

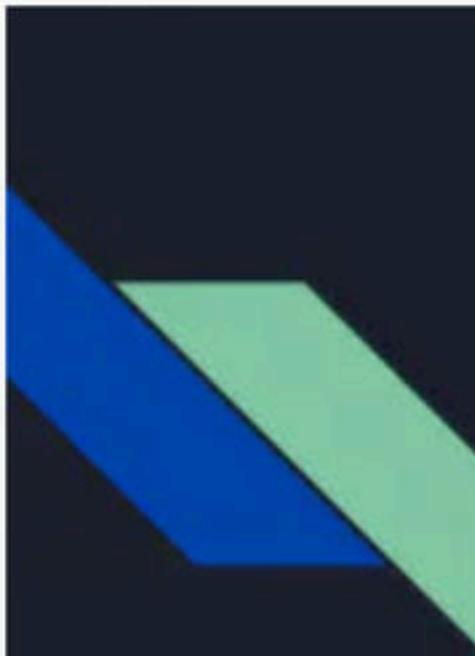


$$(a, b) \Rightarrow (1, 2)$$

$$(1, 2, 4, 5)$$

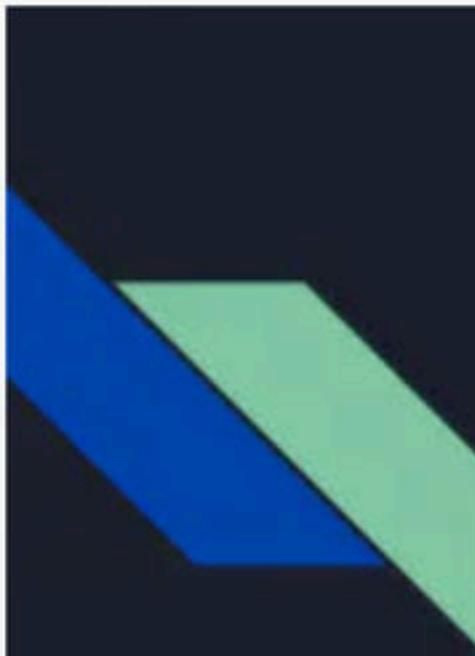
$$\begin{aligned} (1, 4) &\Rightarrow \text{circle with } 2 \\ (1, 3) &\Rightarrow \underline{-5} \end{aligned}$$

$$(c, d) \Rightarrow (4, 5)$$



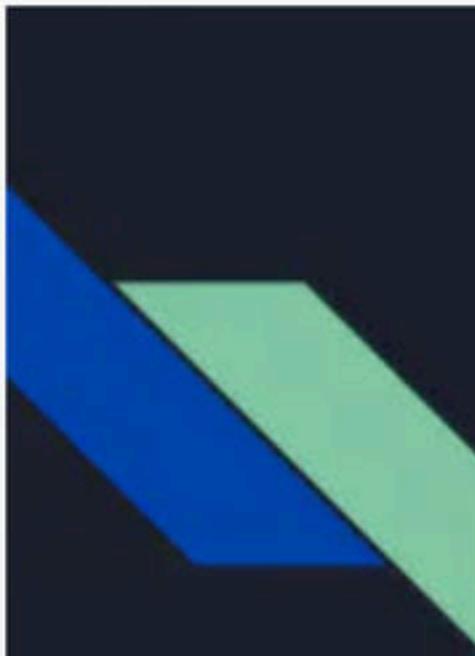
N = 6

A = [2, 3, 1, -4, -1, 30]



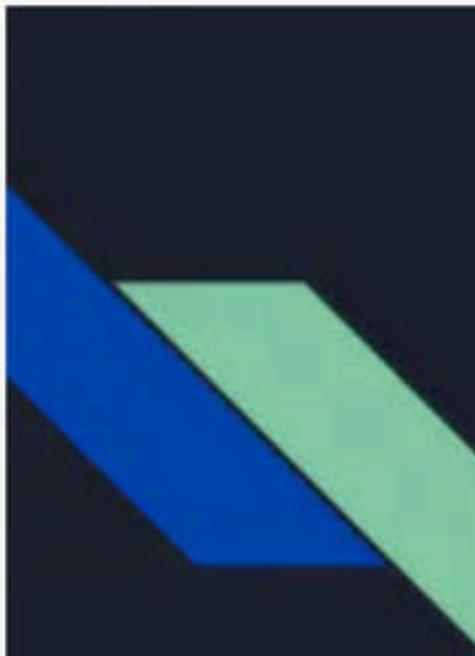
N = 6

A = [2, 3, 1, -4, -1, 30]



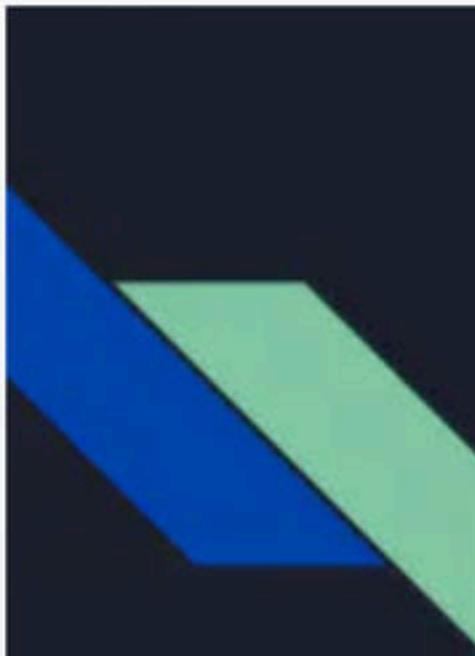
N = 6

A = [2, 3, 1, -4, -1, 30]



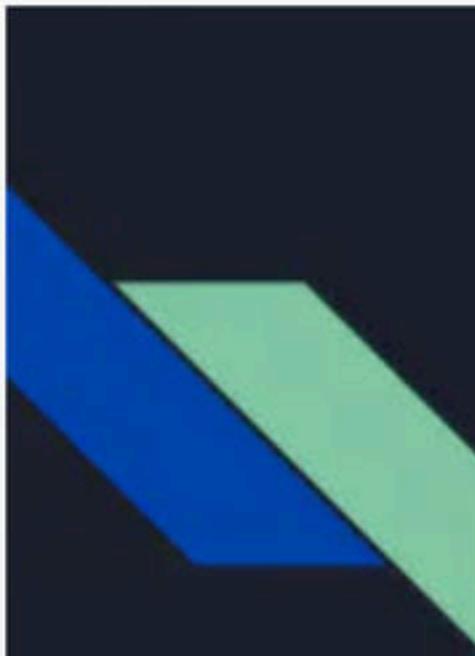
N = 6

A = [2, 3, 1, -4, -1, 30]



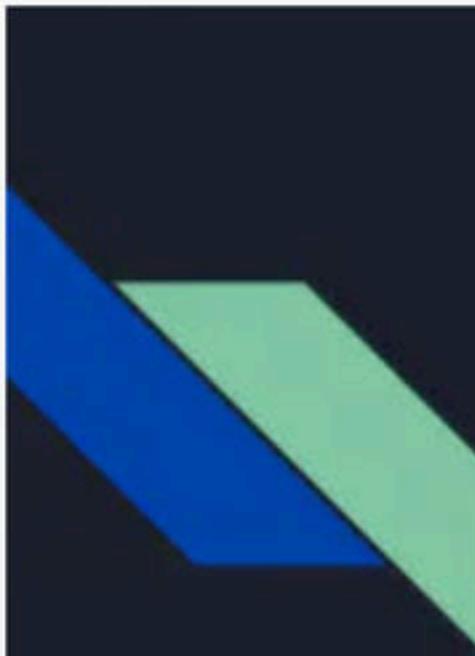
N = 6

A = [2, 3, 1, -4, -1, 30]



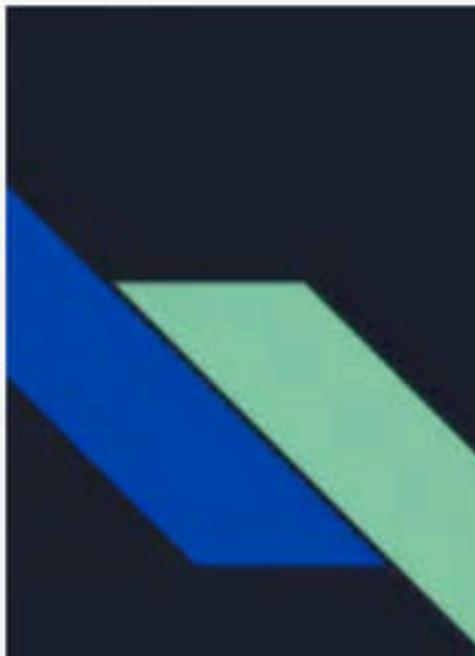
N = 6

A = [2, 3, 1, -4, -1, 30]



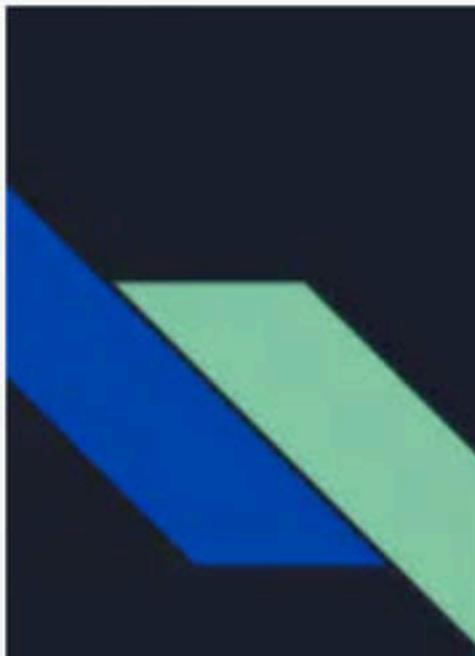
N = 6

A = [2, 3, 1, -4, -1, 30]



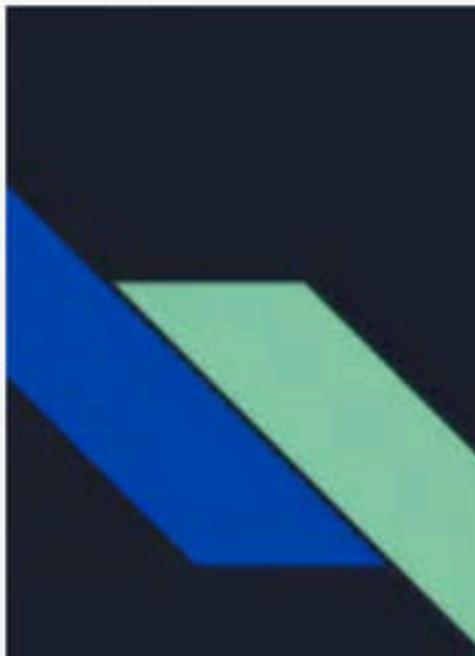
N = 6

A = [2, 3, 1, -4, -1, 30]



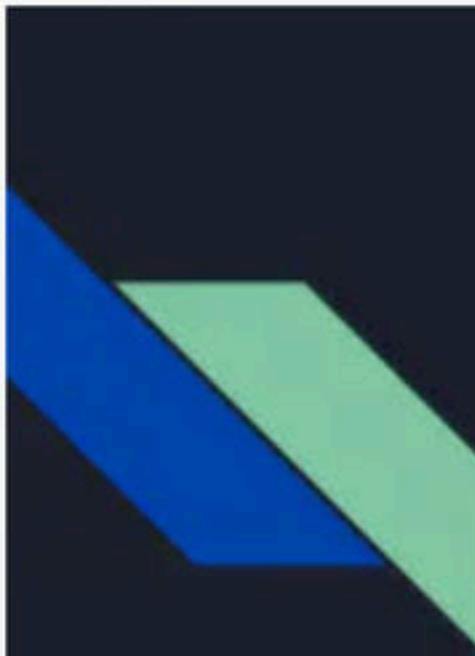
N = 6

A = [2, 3, 1, -4, -1, 30]



N = 6

A = [2, 3, 1, -4, -1, 30]



N = 6

A = [2, 3, 1, -4, -1, 30]

```
bool solve(int A[], int N) {
    set<int> mySet;
    for (int a = 1; a <= N; a++) {
        for (int b = 1; b <= N; b++) {
            int leftSum = A[a] + A[b];
            mySet.insert(leftSum);
        }
    }

    for (int c = 1; c <= N; c++) {
        for (int d = 1; d <= N; d++) {
            int rightSum = A[c] + A[d];
            if (mySet.find(-rightSum) != mySet.end()) {
                return true;
            }
        }
    }
    return false;
}
```

left

right

```
bool solve(int A[], int N) {
    set<int> mySet;
    for (int a = 1; a <= N; a++) {
        for (int b = 1; b <= N; b++) {
            int leftSum = A[a] + A[b];
            mySet.insert(leftSum);
        }
    }

    for (int c = 1; c <= N; c++) {
        for (int d = 1; d <= N; d++) {
            int rightSum = A[c] + A[d];
            if (mySet.find(-rightSum) != mySet.end()) {
                return true;
            }
        }
    }
    return false;
}
```

```
bool solve(int A[], int N) {
    set<int> mySet;
    for (int a = 1; a <= N; a++) {
        for (int b = 1; b <= N; b++) {
            int leftSum = A[a] + A[b];
            mySet.insert(leftSum);
        }
    }

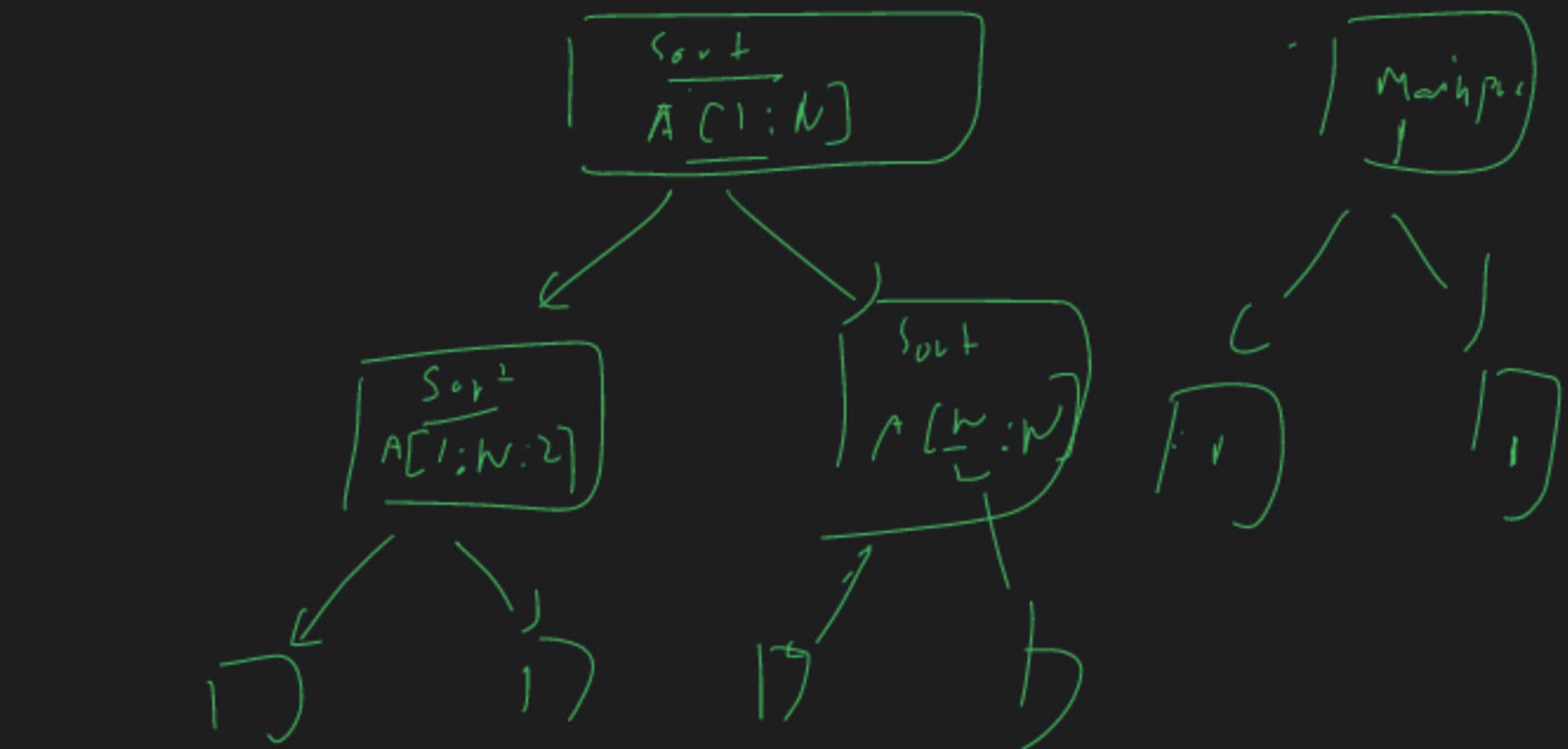
    for (int c = 1; c <= N; c++) {
        for (int d = 1; d <= N; d++) {
            int rightSum = A[c] + A[d];
            if (mySet.find(-rightSum) != mySet.end()) {
                return true;
            }
        }
    }
    return false;
}
```

Assume the time-complexity of set.insert and set.find are  $O(\log N)$ , what will the be the time complexity of solution?

- A.  $O(N^4)$
- B.  $O(N^2)$
- C.  $O(N^2 * \log N)$
- D.  $O(N^3)$

```
bool solve(int A[], int N) {  
    set<int> mySet;  
    for (int a = 1; a <= N; a++) {  
        for (int b = 1; b <= N; b++) {  
            int leftSum = A[a] + A[b];  
            mySet.insert(leftSum);  
        }  
    }  
  
    for (int c = 1; c <= N; c++) {  
        for (int d = 1; d <= N; d++) {  
            int rightSum = A[c] + A[d];  
            if (mySet.find(-rightSum) != mySet.end()) {  
                return true;  
            }  
        }  
    }  
    return false;  
}
```

$O(N^2 \cdot \log N)$



You are given an array **A** of size **N** and an integer **X**.  
Find out if there exists any subset in **A** with a sum of **X**.

Consider the sum of the empty subset to be **0**.

**Constraints:**

$1 \leq N \leq 32$

$-10^9 \leq A_i, X \leq 10^9$

$(a, b) \Rightarrow 3$

$(c, d) \Rightarrow -3$

pair problem

problem

problems

$I_{(c, d)}$

$c, d$

$(r, b, od)$   
 $A[r] + A[s]$

$y_c$

No  
 $\frac{(a+b)}{=0}$

$\frac{(a)}{=0}$  No

$$N = 4$$

$$X = 2$$

$$A = [1, 3, -1, -2]$$

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & -1 & -2 \end{bmatrix}$$

yes

$$\{1, 3, -1\} \Rightarrow 3+2$$

$$\{3, -1\} \Rightarrow 2=2$$

N = 5

X = 11

A = [1, 3, 2, 6, 0]

$$A = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 3/2 & 6 \end{bmatrix} \xrightarrow{?}$$

$$\{4, 3, 2, 6\} \Rightarrow 12$$

$$\{3, 2, 6\} \Rightarrow 11 - 1$$

Yes

N = 5

X = -6

A = [2, 3, 1, -4, -1]

- 4 - 1  $\Rightarrow$  

What will be the answer?

N = 6

X = 11

A = [1, 3, 2, 6, 0, 13]

```
for (each subset S of A) {  
    if (sum(S) == X) {  
        return Yes;  
    }  
}  
return No;
```

N = 6

X = 11

A = [1, 3, 2, 6, 0, 13]

A =  $\begin{array}{c} \swarrow \quad | \quad \searrow \\ \boxed{1 \ 2 \ 5} \end{array}$

S = {3}

S = {0}

S = {13}

{1, 0}

{2}

{2, 0}

{1, 1}

{2, 1, 0}

$\frac{2 + 16}{6}$

0 0 0 0 0 0  
0 0 1 0 1 1  
0 1 6 0 6 2  
0 1 / 3 1 3  
1. 0 0 0 0 4  
1 0 1 0 1 5  
0 1 0 1 0 6  
1 0 1 1 0 7

$N = 6$

$X = 11$

$A = [1, 3, 2, 6, 0, 13]$

$3 \{$

$O(2^n \times n)$

$\textcircled{1011} \quad n$

```
for (mask <= 0; mask < 2^n; mask++) {  
    for (bit = 0; bit < n; bit++) {  
        if (bit is set in mask)  
            result = A[bit]  
        if (result == X)  
            return Yes;  
    }  
}
```

return No;

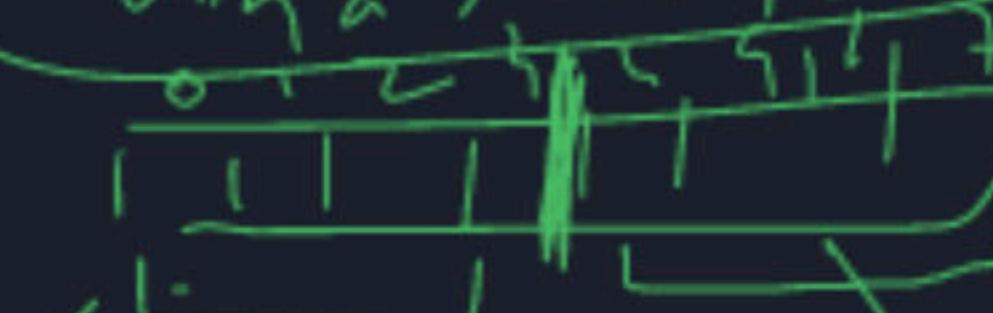
$$N = 6$$

$$X = 11$$

$$A = [1, 3, 2, 6, 0, 13]$$

Main problem

Find if there  
ex. is  $\exists$  a subset  $S$   
with a sum of  $X$



$S = \{ \underbrace{(0, 1)}, \underbrace{(1, 5)}_{\text{sum}} \}$

$$\left\langle \frac{N}{2} \right\rangle = \frac{N}{2}$$

$A[\frac{N}{2}:N]$

Find sums  
for all subsets  
of  $A[\frac{N}{2}:N]$

$A[0:\frac{N}{2}]$

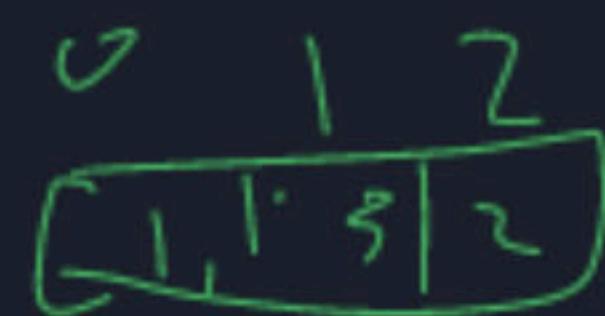
Find sums  
all possible  
subsets of

$A[0:\frac{N}{2}]$

N = 6

X = 11

```
A = [1, 3, 2, 6, 0, 13]
```



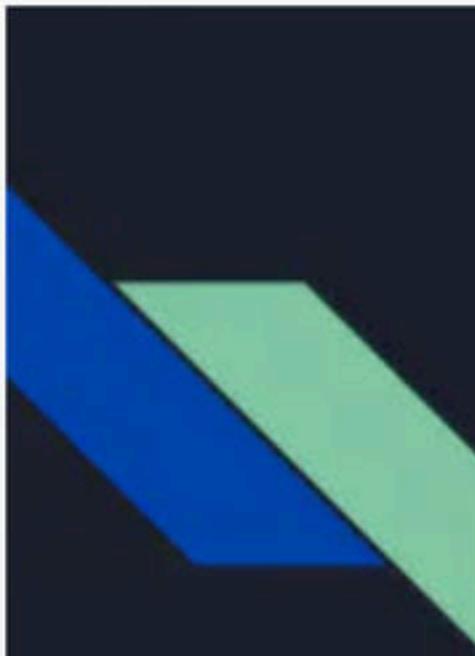
1

$$\underline{\{3, 2\}}$$

七

~~rights~~

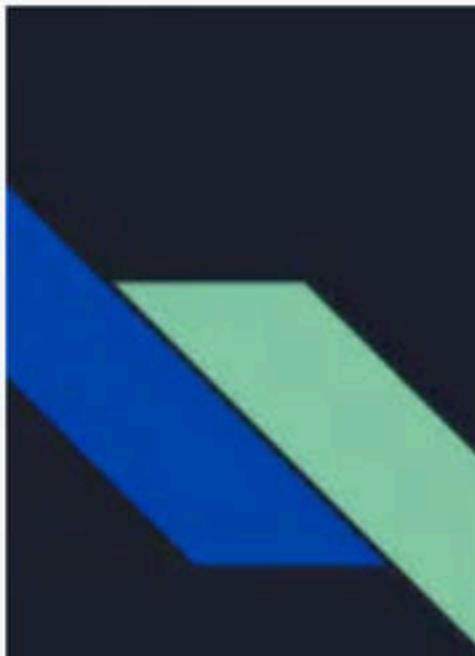
b 1 2



N = 6

X = 11

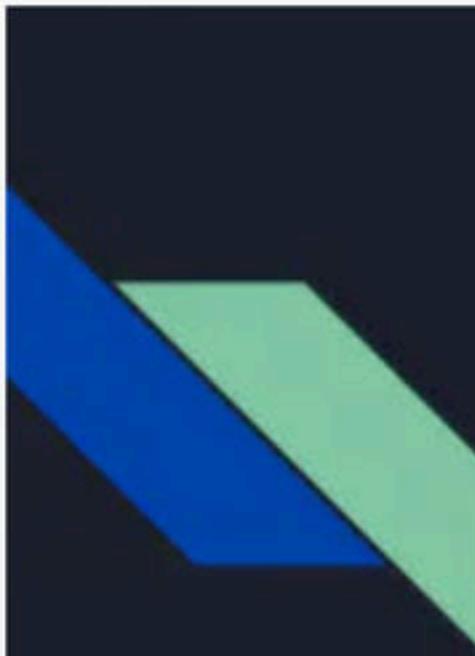
A = [1, 3, 2, 6, 0, 13]



N = 6

X = 11

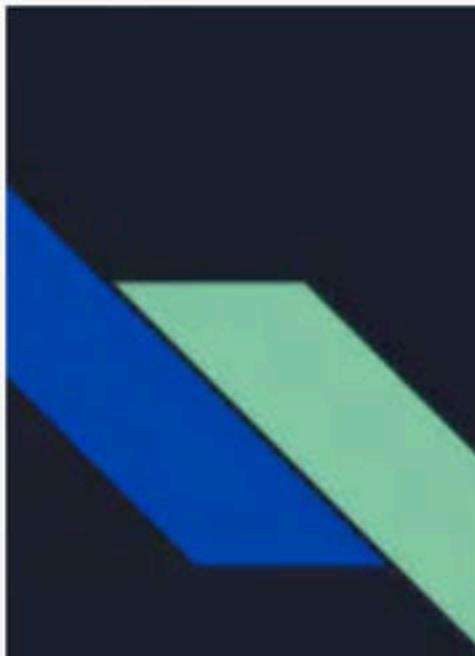
A = [1, 3, 2, 6, 0, 13]



N = 6

X = 11

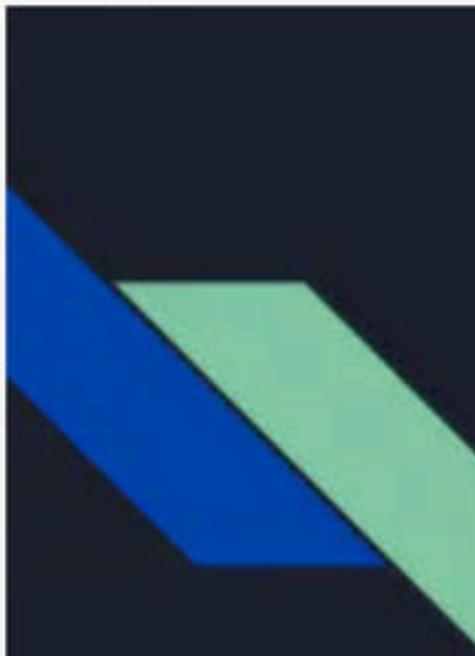
A = [1, 3, 2, 6, 0, 13]



N = 6

X = 11

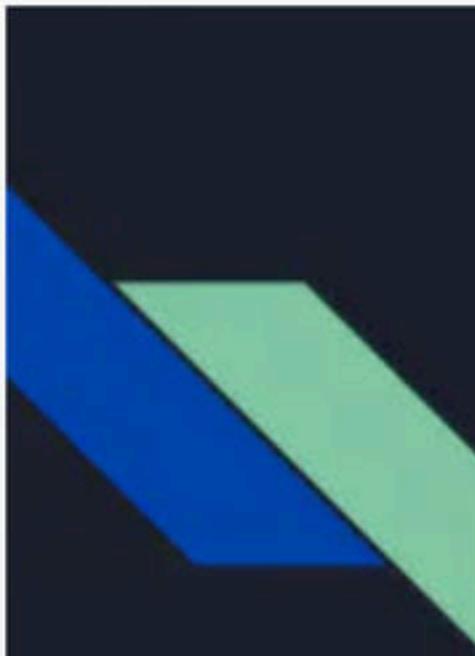
A = [1, 3, 2, 6, 0, 13]



N = 6

X = 11

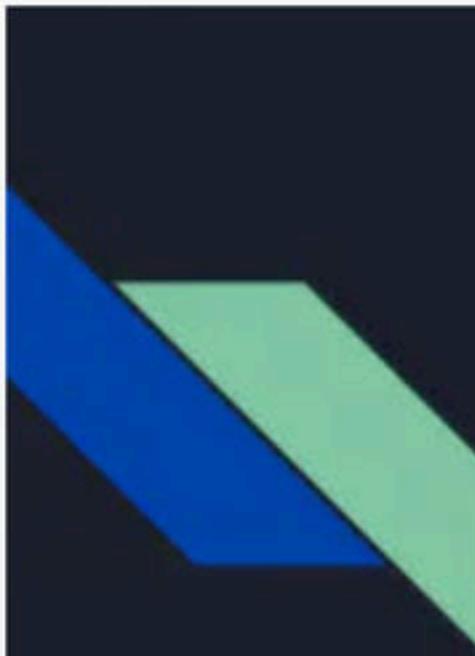
A = [1, 3, 2, 6, 0, 13]



N = 6

X = 11

A = [1, 3, 2, 6, 0, 13]



N = 6

X = 11

A = [1, 3, 2, 6, 0, 13]

```
bool solve(int A[], int N,
    if (X == 0) {
        return true;
    }
    if (N == 1) {
        return A[0] == X;
    }
    int leftSZ = N / 2;
    int rightSZ = N - leftSZ;
    int left[leftSZ + 5];
    int right[rightSZ + 5];
    { for (int i = 0; i < leftSZ; i++)
        left[i] = A[i];
    }
    { for (int i = leftSZ; i < N; i++)
        right[i - leftSZ] = A[i];
    }
    set<int> mySet;
    for (int i = 0; i < (1 << N) - 1; i++) {
        int leftSum = 0;
        for (int j = 0; j < leftSZ; j++)
            if (((i >> j) & 1) == 1)
                leftSum += left[j];
        mySet.insert(leftSum);
    }
}
```

```

for (int i = 0; i < (1 << rightSZ); i++) { O(2^M/2)
    int rightSum = 0;
    for (int j = 0; j < rightSZ; j++) {
        if ((i >> j) & 1) {
            rightSum += right[j];
        }
    }
    int required = x - rightSum;
    if (mySet.find(required) != mySet.end()) {
        return true;
    }
}
return false;

```

$O(2^{M/2} \times h + O(2^M))$

```

bool solve(int A[], int N, int X) {
    if (X == 0) {
        return true;
    }
    if (N == 1) {
        return A[0] == X;
    }

    int leftSZ = N / 2;
    int rightSZ = N - leftSZ;
    int left[leftSZ + 5];
    int right[rightSZ + 5];

    for (int i = 0; i < leftSZ; i++) {
        left[i] = A[i];
    }
    for (int i = leftSZ; i < N; i++) {
        right[i - leftSZ] = A[i];
    }

    set<int> mySet;
    for (int i = 0; i < (1 << leftSZ); i++) {
        int leftSum = 0;
        for (int j = 0; j < leftSZ; j++) {
            if ((i >> j) & 1) {
                leftSum += left[j];
            }
        }
        mySet.insert(leftSum);
    }
}

```

```

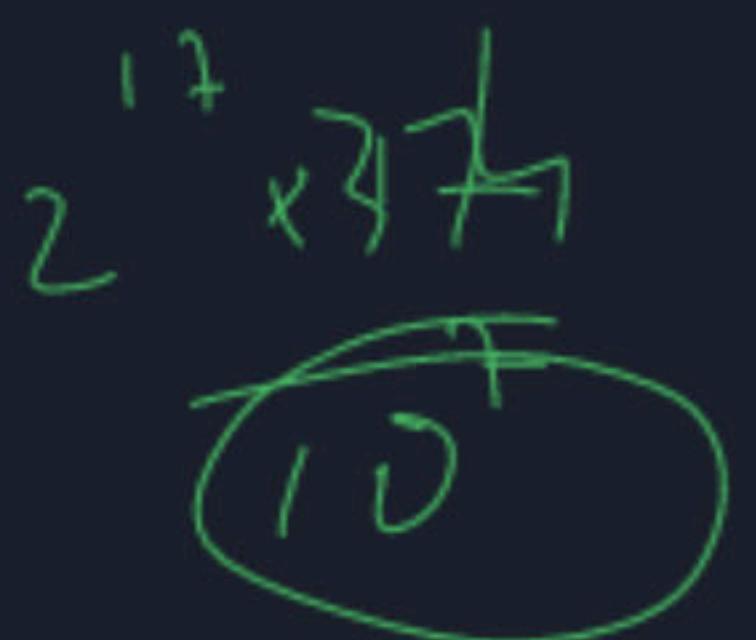
        for (int i = 0; i < (1 << rightSZ); i++) {
            int rightSum = 0;
            for (int j = 0; j < rightSZ; j++) {
                if ((i >> j) & 1) {
                    rightSum += right[j];
                }
            }
            int required = X - rightSum;
            if (mySet.find(required) != mySet.end()) {
                return true;
            }
        }
        return false;
    }
}

```

$O(2^n)$

$\Rightarrow O(2^{n/2})$

$2^{n/2} \times n$



```
bool solve(int A[], int N, int X) {
    if (X == 0) {
        return true;
    }
    if (N == 1) {
        return A[0] == X;
    }

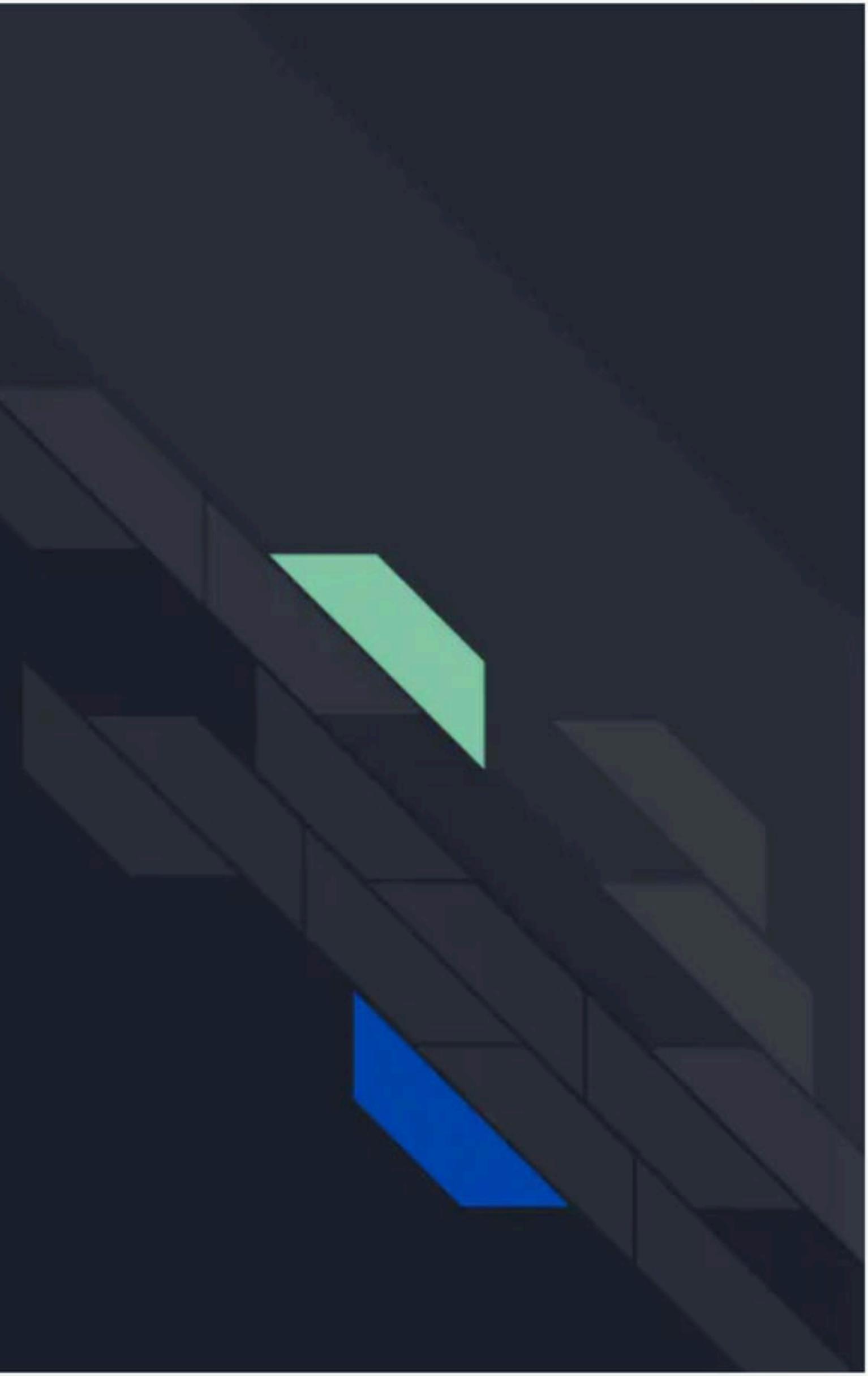
    int leftSZ = N / 2;
    int rightSZ = N - leftSZ;
    int left[leftSZ + 5];
    int right[rightSZ + 5];

    for (int i = 0; i < leftSZ; i++) {
        left[i] = A[i];
    }
    for (int i = leftSZ; i < N; i++) {
        right[i - leftSZ] = A[i];
    }

    set<int> mySet;
    for (int i = 0; i < (1 << leftSZ); i++) {
        int leftSum = 0;
        for (int j = 0; j < leftSZ; j++) {
            if ((i >> j) & 1) {
                leftSum += left[j];
            }
        }
        mySet.insert(leftSum);
    }
}
```

```
for (int i = 0; i < (1 << rightSZ); i++) {
    int rightSum = 0;
    for (int j = 0; j < rightSZ; j++) {
        if ((i >> j) & 1) {
            rightSum += right[j];
        }
    }
    int required = X - rightSum;
    if (mySet.find(required) != mySet.end()) {
        return true;
    }
}
return false;
}
```

# Summary



You are given an array  $A$  of size  $2^N$ .

For each  $i$  ( $0 \leq i \leq 2^N - 1$ ) you need to find  $\sum A_j$  where  $i \& j = j$ .