

- 31-oct → Leet Code ~~पर्याप्त नहीं~~ तो कौन सी बारे लिए जाएंगे
- (Dec 8 exam के तारीख 8 Nov के तारीख) prepare ~~हो~~ इसके लिए

Rajiv Singh.

Content	Revision - Date
①	HashMap, heap
②	stack.
③	Recursion.
④	LL
⑤	Tree.

Types of DP (optimization over plain rec).

E-M.

- ① Knapack → ① 0-1
② Bounded
③ Unbounded
 - ② LCS (longest common subsequence)
 - ③ LIS (longest increasing subsequence)
 - ④ Matrix chain multiplication
 - ⑤ dp on grid
 - ⑥ Kadane's algorithm
 - ⑦ others
- } Companies Interview
17)

Advanced dp - (CP)

- ① DP k-th - lexicographical String
- ② dp on tree
- ③ dp + Bitmasking
- ④ dp + Bit / segment tree
- ⑤ DP + convolution
- ⑥ dp - pre processing
- ⑦ dp + trie
- ⑧ dp + geometry
- ⑨ dp + B.S.
- ⑩ dp + knuth optimization

Lecture-2:HashMap & heap.

Sorting of array using do heap.

int arr[] = new int { };

- ① O(n) space.
- ② $n \log n$ time complexity
- ↓
 downheapify
 for element x

Step:

- ① convert arr to heap & maxheap.
- ② swap Max & last.
- ③ Call maxheap.
- ④ Repeat all steps.

Ex. int arr[] = {10, 20, 30, -2, -3, -4, 5, 6, 7, 8, 9, 22, 11, 13}.

- ① heap sort easy & time is also,

$n \rightarrow$ if heap sort

$$+ n \log n \text{ sort } \rightarrow = n + n \log n.$$

Benefits:-Merge, quick \Rightarrow (B.S.)Easy, fast and easy to track mistakes.

Qn Recursion :- how to count space.

- ② Algo. :- ① Array \rightarrow heap. (downheapify) Required.
- ② while ($Pq.size() \neq 0$) {

```
swap(arr[0], arr[i--]); // Pq.pop();
// Virtually reduce size
```

Qn k^{th} largest element

- ① Brute Force = $n \cdot k = (k=n)$.
- ② sorted array ($n \log n$) $\xrightarrow{\text{arr}[n-k]}$
- ③ using lb.

Algo. 1. insert first k elements.

(2)

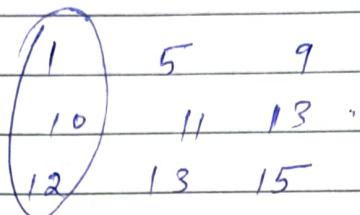
703, 378.

① Comparator &

Comparable

one ✓ 703

✓ 378



②

$$\begin{aligned} 1 \rightarrow 5 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 13 \\ \rightarrow 13 \rightarrow 20. \end{aligned}$$


A java का ना भी class का लोग वह क्या है

① add → implement → pick child compare (Integers)

but how it compare student ?

So, pg. wants a GT class bhej rhe hain of Comparable interface implement करते हैं।

② interface → Contract of Function जैसा है

③ when class implements interface it provides body of all function of interface.

h

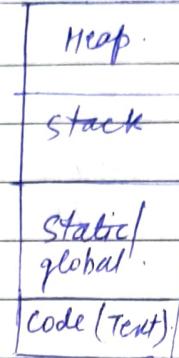
(comparator already written)



④ Pair ↘

973 k closest point

fn int CompareTo (student o) {
 this part on which
 it is called
 this > o +ve}



includ < stdio.h >
includ < stdlib.h >

int main() {

 int a ; // goes on stack.
 int * p ;

 p = (int *) malloc(sizeof(int));
 *p = 10 ;
 free(p);

 p = (int *). malloc(20 * sizeof(int));

}

P-2 int total ;

int square(int x) {

 return x*x ;

}

```
int SqOfSum (int x, int y) {
```

```
    int z = square (x+y);  
    return z; //  $(x+y)^2$   
}
```

```
int main () {
```

```
    int a = 4; int b = 8;
```

```
    total = SqOfSum (a, b);
```

```
    printf (" O/P = %.d", total);
```

891, a.

SOS ()

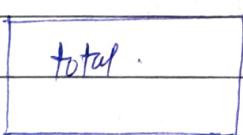
x, y, z.

main ()

a, b

} stack-frame.

Stack diag.



when stack memory goes full
gives stackoverflow.

* Mem. of stack not grow during
Runtime. (Fixed).

* Further allocation and deallo-
cation happen by rule.

when fun call → push on the
stack

and when finished poped from stack

Heap

(dynamic Memory)

(large free pool of
Memory)Stack

To use dynamic Mem. in c. or heap Memory.

- ① malloc
 - ② calloc
 - ③ realloc
 - ④ free
- } fn.

For C++:

- ① new.
 - ② delete.
- } operators

In ~~c~~ c :

main() {

int a; // goes on stack.

int *p;

p = (int*) malloc (sizeof(int));

Return INT & void pointer so type casted to integer:

malloc :- Malloc fn look free space in heap book it and give back the pointer.

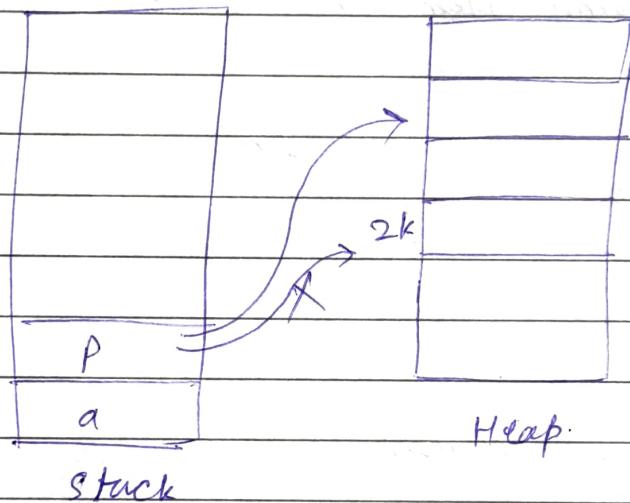
freud).

if we write

$P = (\text{int} *) \text{ malloc}(\text{sizeof}(\text{int}))$;
 $*P = 20;$

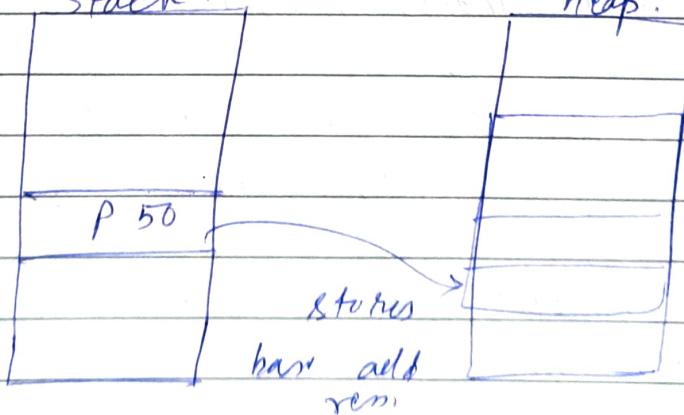
Now it reference to new add let's say 400

so, before pointing it to new address. free because resource is utilized.



For storage of array;

$P = (\text{int}*) \text{malloc}(20 * \text{sizeof}(\text{int}))$;
Stack : _____ Heap : _____



in CPP.

int main() {

int a; // goes on stack
int *p;

p = new int;

*p = 10;

delete p;

p = new int[20];

delete [] p;

→ [delete or free memory]

(Pep - 3).

* To make array

$$\begin{array}{c} [-10 \rightarrow +10] \\ \hline +10 & +10 \\ [0, 20] \end{array}$$

have done shifting of origin.

$$(10^6) \xrightarrow{\text{group}} 10 \text{ (elements.)}$$

* Space \times time = constant +.

$$\boxed{\text{Space or } \frac{1}{\text{Time}}}.$$

e.g. Our formula works on modulus.

$(34, 44, 54, \dots, 84) \rightarrow \textcircled{4}$
 ↑ (collision).

To remove this comes.

l.f. \rightarrow 1 group if max to max
 from element 3rd until 1.

$$l.f. = \frac{(\text{grp size})}{(\text{arr. size})}.$$

load factor

$\ell \neq 1 \rightarrow o(n)$ searching (collision) | 1

$\ell = 1 \rightarrow$ searching $o(n)$

so have to make optimal formula

freq map

Amortized to $o(1)$. not perfect $o(1)$.

freqMap

12ab ABCD 45 \rightarrow (256)

① put(key, value)

② get(key) \rightarrow null (if not found)
return value.

③

null + 1 \rightarrow null pointer exception.

* foreach loop only use for get

* don't use for other \rightarrow gives current error.

1 → concept (hm)
2 → indexing hm
3 →
4 →

APCO
Date : _____
Page : _____

Q. 20

a, b, c, a, c, h, a.
0 1 2 3 4 5 6.

b → 3

7 → 4

a → 0, 3, 6.

8 → 5

b = 1, 5

9 → 6;

c → 2, 4

10 → 7

11 → 8

12 → 9

13 → 10

14 → 11

15 → 12

16 → ans

key	val
0	a
1	b
2	c
3	a
4	c
5	b
6	a

Q. 349, 350, 219, 451, 347, 380.

S. P8V freqMap -> put (String str) {

HashMap < character, ArrayList < Integer >> map =
new HashMap <>();

for (int i=0; i< str.length(); i++) {

char ch = str.charAt(i);

map.put(ch, new ArrayList <>());

map.get(ch).add(i);

3.

3.

key	value	AL ↗							
a	4k								
b	5k								
c	6k								
d	7k								

Qn // 349 Leet Code -

$\{1, 2, 2, 1\} \cap \{2, 2\}$

$\{2, 2\} \Rightarrow \{2\}$

HashSet < Integer > hs = {1, 2}.

* ~~PS~~ variable \Rightarrow 349 set create ~~set~~ \Rightarrow called
hashset;

public int[] intersection (int[] nums1, int[] nums2) {

 HashSet < Integer > map = new HashSet < > ();

 for (int ele : nums1) map.add (ele);

 ArrayList < Integer > ans = new ArrayList < > ();

```

    for (int ele : nums2)
        if (map.contains (ele)) {
            ans.add (ele);
            map.remove (ele);
        }
    }
```

```

int arr[] = new int [ans.size()];
int i = 0;
for(int ele: ans) arr[i++] = ele;

return arr;
}

```

- algo
- ① num1 → HashSet & STC1
 - ② am other at al STC2
 - ③ num2 & element check PTC
if not hs &

Ques 350 LeetCode

- ④ if present then add to al.
- ⑤ al → arr → return arr.

nums1[] = [1, 2, 2, 1], nums2[] = [2, 2].

O/P = [2, 2].

Public int[] intersect (int[] num1, int num2) {

HashMap<Integer, Integer> map = new HashMap<>();

For (int ele: num1) map.put (ele, map.getOrDefault (ele, 0) + 1);

ArrayList<Integer> ans = new ArrayList<>();

For (int ele: num2) {

if (map.containsKey (ele)) {

ans.add (ele);

map.put (ele, map.get (ele) - 1);

if (map.get (ele) == 0) map.remove (ele);

}

?

```
int[] arr = new int[arr.length];
int i = 0;
```

```
for (int ele : arr) arr[i++] = ele;
```

```
return arr
```

```
}
```

Q4

219 contain duplicate II

(LeetCode)

arr → Int

Int - K

2 repeated after ab [index of diff ≤ K]

Q5 Public boolean containsNearbyDuplicate (int[] nums,
int k) {

```
HashMap<Integer, Integer> map = new HashMap<>();
```

```
for (int i = 0; i < nums.length; i++) {
```

```
if (map.containsKey(nums[i])) {
```

```
int index = map.get(nums[i]);
```

```
if (i - index) ≤ k) return true;
```

```
}
```

```
map.put (nums[i], i);
```

```
return false;
```

```
}
```

key val.

1	0
2	1
3	2
1	3
2	4
3	5

$$(3-0) < 2$$

Qw

11/22/20 Leetcode -

[1, 2, 3, 1]

key	Value.
1	0
2	1
3	2
1	3.

Qn 45 Sort character by freq.

StringBuilder sb

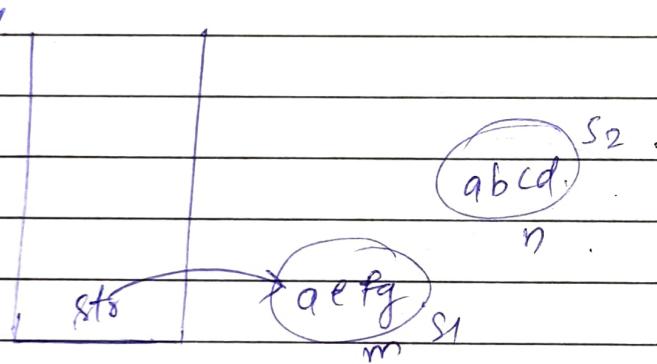
use to add something in string

n, m

\uparrow $O(n+m)$ other concat operation

but if we sb. $\rightarrow O(m)$.

str = "aefg".



$$\text{str} = \text{str} + \text{s2}$$

$$= O(m+n).$$

String s = " ";

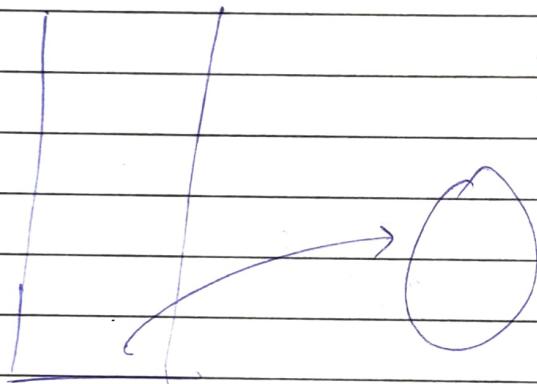
for int i = 0 ; i <

sb \rightarrow used inside the programme.

Q11 347

$[1, 1, 1, 2, 2, 3]$, $k = 2$.

key	
1	1, 2, 3
2	2, 2
3	

Q11 380 451

public String frequencySort (String str) {

 HashMap< Integer, Integer > map = new HashMap<>();

 for (int i=0; i<str.length(); i++) {

 char ch = str.charAt(i);

 map.put(ch, map.getOrDefault(ch, 0)+1);

}

Priority Queue < character > pq = new Priority Queue<(a,b)
} } { .

return map.get(b) - map.get(a); } } .

for(character ch: map.keySet()) {
pq.add(ch);
} .

StringBuilder sb = new StringBuilder();

while(pq.size() != 0) {

character ch = pq.remove();
int freq = map.get(ch);

for(int i=0; i < freq; i++) sb.append(ch);

}

return sb.toString();

(380) PW

sprinkler -

① longest Palindrome.

②

③ Tries (Huffman Coding) → adv. ds - basic

④ Resume based qn.

⑤ HR Round Personality

① Situation based.

① ds algo.

January

② Project

Comparator.

#

class Student {

int rno;

int ht;

int wt;

student (int rno, int ht, int wt) {

this.rno = rno;

this.ht = ht;

this.wt = wt;

}.

public int compareTo (Student o) {

return this.rno - o.rno;

}.

Public int compare () {
strong testing
}

return "Rno = " + this.rno + ", Ht = " + this.ht
+ ", wt = " + this.wt;

{

M-2 Using Comparator

static class StudentHt Compt implements Comparator<
Student> {

Public int compare (Student s1, Student s2) {

return s1.ht - s2.ht;

{

{

static class StudentWt Compt implements
Comparator <Student> {

Public int Compare (Student s1, Student s2) {

return s1.wt - s2.wt;

{

{

Similarly, we can write Comparator to compare any other objects element.

* how to use :-

① if Priority Queue < Student > PQ = new Priority Queue();

will run on the basis of Comparable.

②

Priority Queue < Student > PQ = new Priority Queue < > /
new Student wt. Compf();

wt. Comparator at object.

Similarly can pass ht Comparator also.

Heap construction -

```
import java.io.*;
import java.util.*;
```

Public class main {

Public static class Priority Queue < T > {

ArrayList< T > data;

Comparator Comp;

Public Priorityqueue () {

 data = new ArrayList < () ;

 comp = null;

}

Public Priorityqueue (Comparator comp) {

 data = new ArrayList < () ;

 this.comp = comp;

}

Private boolean isSmaller (int i, int j) {

 if (comp == null) {

 comparable ith = (Comparable) data.get (i);

 " jth " = (Comparable) data.get (j);

 if (ith.comparable (jth) < 0) {

 return true;

 }

 } else {

 T ith = data.get (i);

 T jth = data.get (j);

 if (comp.comparable (ith, jth) < 0) {

 return true;

 } else {

 return true; }

Public Priority Queue () {

data = new ArrayList < > ();

comp = null;

}

Public Priority Queue (Comparator comp) {

data = new ArrayList < > ();

this.comp = comp;

}

Private boolean isSmaller (int i, int j) {

if (comp == null) {

comparable ith = (comparable) data.get(i);

" jth = (comparable) data.get(j);

if (ith.comparable(jth) < 0) {

return true;

}

else {

return false;

}

else {

T ith = data.get(i);

T jth = data.get(j);

if (comp.compare(ith, jth) < 0) {

return true;

else {

return false; }

- ① Comparable
- ② Comparator
- ③ how it work in java pg.
- ④ how it works in our pg.

$c.rno \leftarrow o.rno$, this \rightarrow $-ve$ value
 ↑ this ↑ parent

Comparator(P)

$-ve$ value \rightarrow this after child after

$E < P$

GT after UT of 342 512

$o.rno - this.rno$, 342 GT of cond.
 $o.rn \neq this.rn$,

$-ve \rightarrow E < P \rightarrow C$ 342 b/w 512

Or

($O > o.rno - c.rno$,

$\rightarrow -ve \rightarrow$

C 342 b/w 512
 $c.rno > o.rno$.

100 0,

$(0,0)$, $(0,1)$, $(0,2)$, $(0,3)$

1	2	3	5
---	---	---	---

$$\frac{A[a[0]]}{0} * \frac{A[b[1]]}{1} - \frac{A[a[1]]}{0} * \frac{A[b[0]]}{0}$$

$$1 * 2 - 1$$

① \rightarrow ② (4th)

2	7	4	1	8	1
0	1	2	3	4	5

for ($n-1 \rightarrow 0$)

$$a = arr[n-1];$$

$$a = arr[n-2];$$

$$\text{Math.abs}(a-b);$$

as ~~for~~

for (int i = n-1; i >= 0; i--) {

$$a = arr[i]; \quad 5 \quad 4 \quad 3, 2, 1$$

$$b = arr[i-1]; \quad 4 \quad 3, 2, 1, 0$$

$$arr[i] = \text{Math.abs}(a-b); \quad \neq$$

$(0,1)$, $(0,2)$, $(0,3)$, $(0,4)$, $(0,5)$.

$$\frac{A[a[0]]}{A[a[1]]} = \frac{B[b[0]]}{B[b[1]]}$$

$$\frac{11}{2} = 0.5 \cdot \frac{1}{3} = 0.3 \Rightarrow 3^{-2}$$

$$(3^{-2})$$

$$3^{-2}$$

$$\begin{array}{r}
 112 \quad 01 \\
 10 \\
 \hline
 11 \rightarrow \\
 \underline{12} = 3
 \end{array}$$

Ques
135

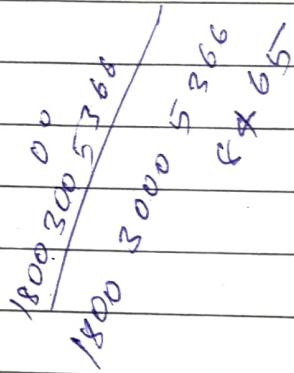
N = length of dic.

k = pattern size.

{abb, abc, xyz, yyz}.

760 →
1 2

xyzzyzzabcdd



mmn ooppl ≠ mm.

x = 1

y = 1

z = 2

w = 3

v = 3.

a = 4

b = 4

c = 5

d = 6.

e = 7.

f = 7

g = 7

Hash Map <

(String and Integer)

~~for (if contains key) <~~

x	1
y	2
z	3.
a	4
b	5
c	6
d	7

(19/08/2020).

11 128.

3

longest consecutive subsequence

10, 11, 9, 103, 2, 5, 100, 101, 104, 105, 102, 8, 12,
1000, 1006, 1007.

Set pos ev and next

Array of ll is used by H.M-

Linked list < Integer > Arr[].

11 49 group anagrams.

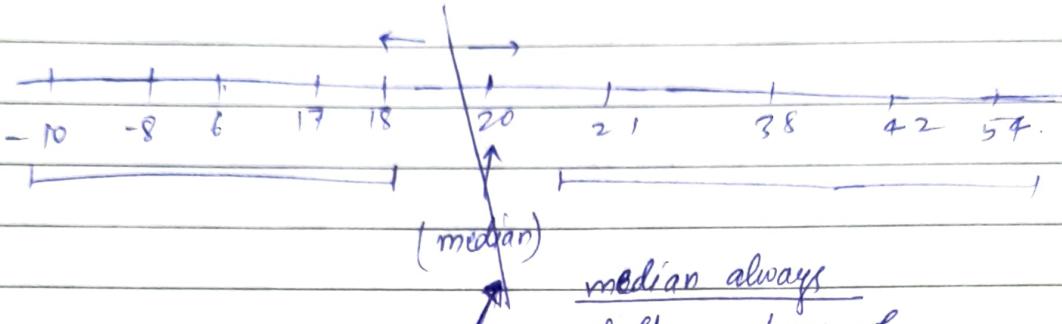
abc abc aaa bbb CCC c

a-5
b-5
c-6. } 2
uniscode.

$$n \left(26 + l + 26 + l + R + R + R \right) \\ = 5nl.$$

II 295

Qn Find median From a stream of integers.

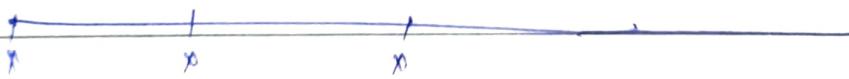


Ans:

Qn Largest Subarrays with 0 and 1

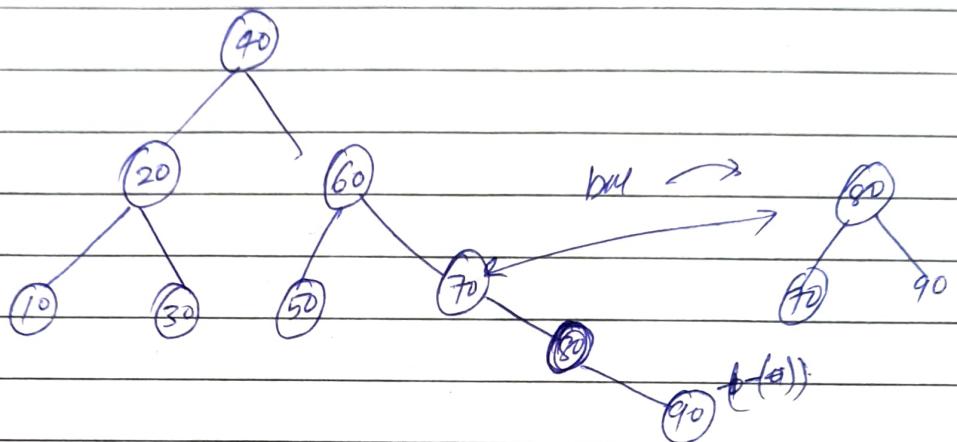
Qn Count Subarrays with equal no. of 0 and 1

Hint



BSI

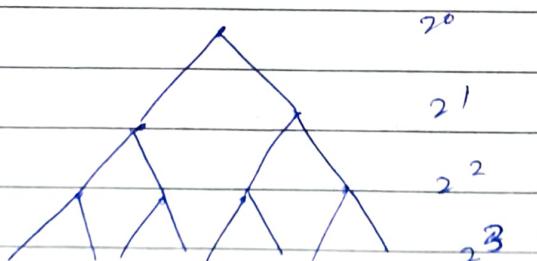
10	20	30	40	50	60	70	80	90
0	1	2	3	4	5	6	7	



$$0 \leq |lh - rh| \leq 2$$

$$-1 \leq (lh - rh) \leq 1$$

Balanced BST :- Have equal no. of child on left and right.



$n = 2^0 + 2^1 + 2^2 + \dots + 2^h$
 total nodes
 or elements

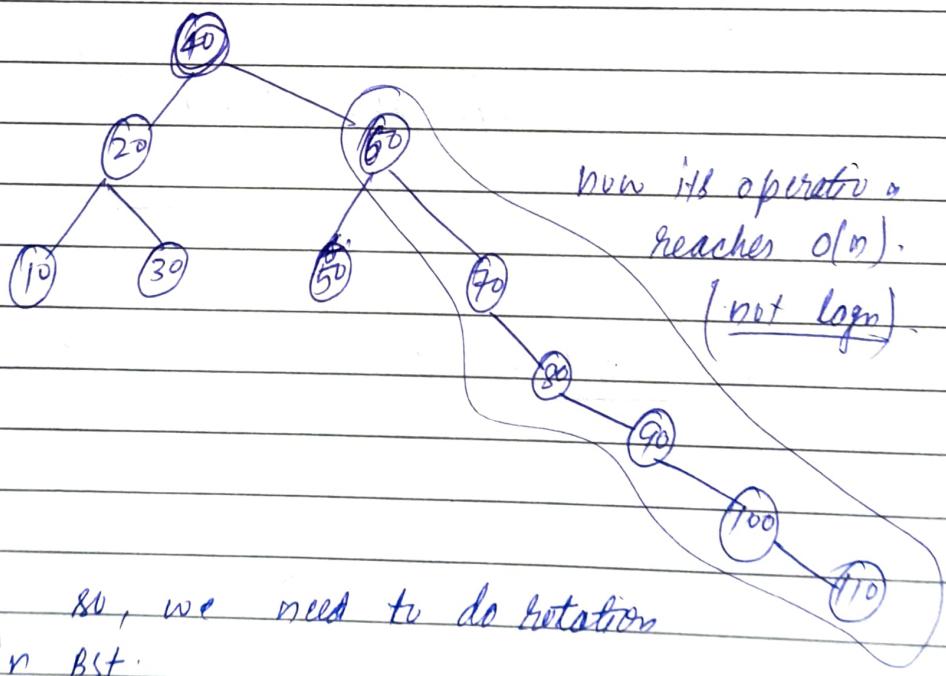
$$n = (1) \frac{(2^h - 1)}{2 - 1}.$$

$$\log_2 N = \log_2^h$$

$$h = \log_2 n$$

- ① searching
 ② insertion
 ③ deletion } op. $\rightarrow \log(n)$. in BST.

which in BT $\rightarrow O(n)$.



So, we need to do rotation
 in BST.