binary search

arr | 2 | 4 | 6 | 8 | 9 | 15 | 20 | 21 |

target 7

arr | 2 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 5 | 6 | 7 |

0 1 2 3 4 5 6 7

target = 4

find first occurence of 4

ans 3

find last occur of 4

ans = 6

find largest no which is smaller than target

2	4	6	20	30	42

target = 9 ans 6

= 28 ans 20

find smallest no which is greater than target

target = 21 ans 30

9 ans 20

28 ans 30

$$\log_a b = \left(\frac{\log_c b}{\log_c a} \right)$$

Math.log10(100) = 2

$$\frac{\text{Math.log}_{10}(1B)}{\text{Math.log}_{10}(\text{maxGrowthRate})}$$

Stack and queue

Finding next greater
smaller

---

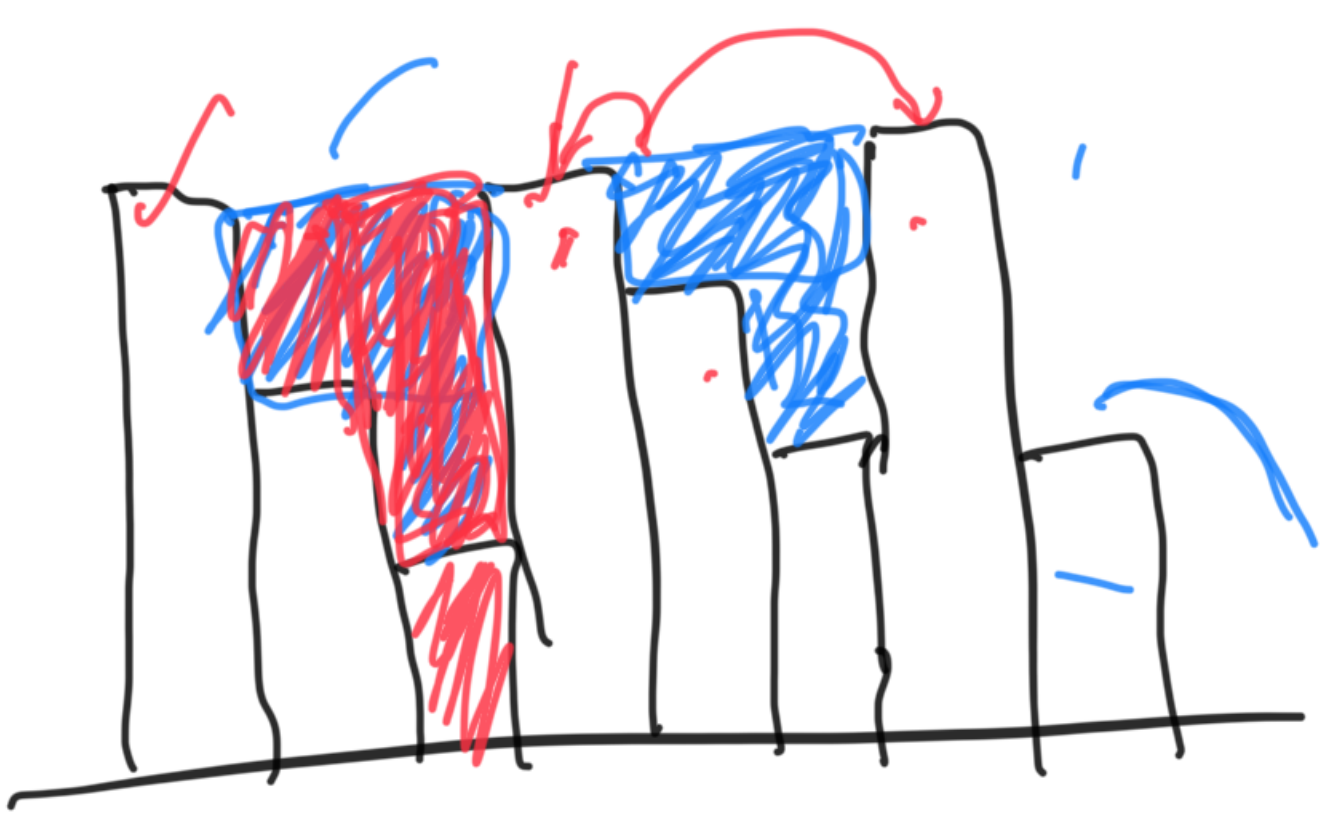
s = [1, 2, 4, 1, 5] → [2, 4, 5, 5, -1]

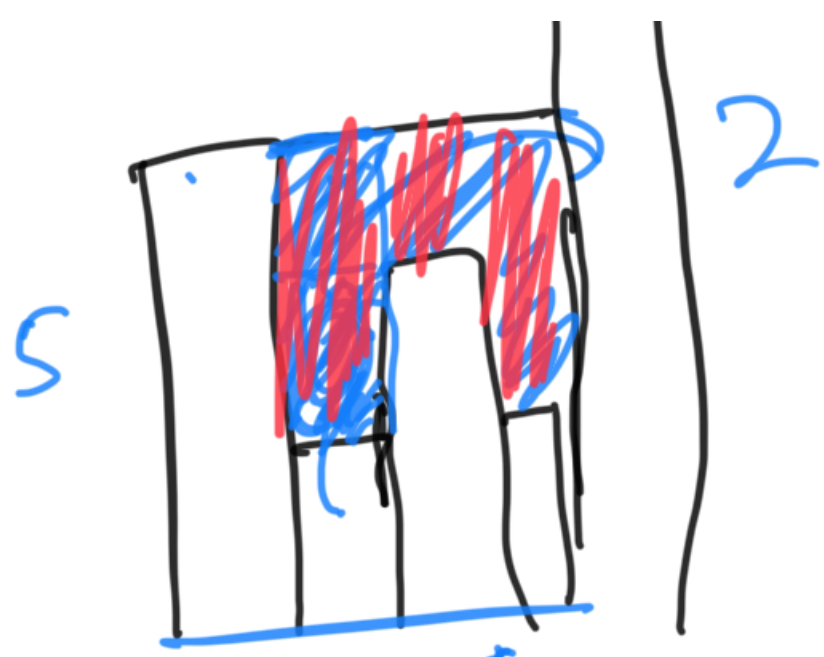
scan

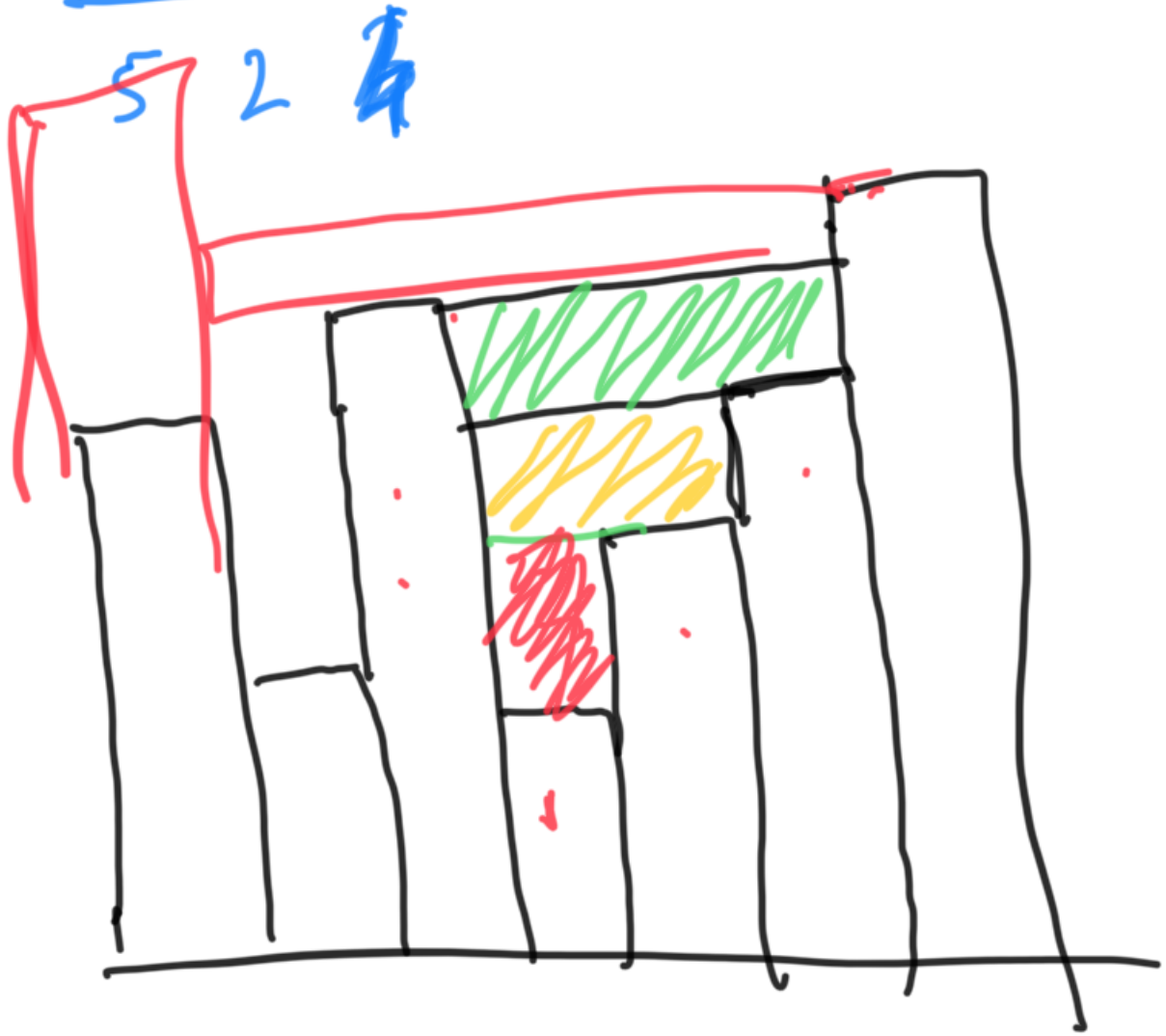
res

1
2
4
1
5

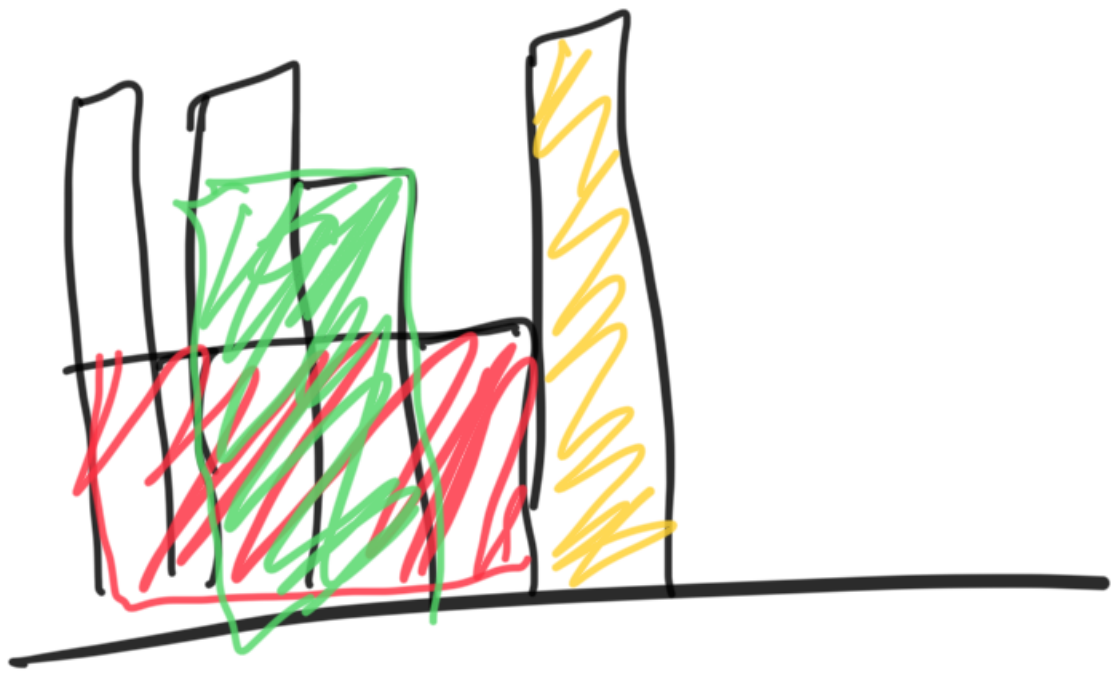
-1,
-1,-1
-1,-1,-1
-1,-1,-1,4
[-1,-1,-1,4,-1







Largest rectangle histogram



K. Kadane's Algo

Max sub array sum

[_ _ _ _]

Algo

cur = 0

max = -INF

for (i=0 ; i< arr.length ; i++)

{

cur += arr[i].

max= max (max, cur)

if(cur < 0)

{ cur = 0

}

}

reset

[-5 , -3, -1, -7]

-1

max=-INF

-5

-5 max = -5

-3 max = -3

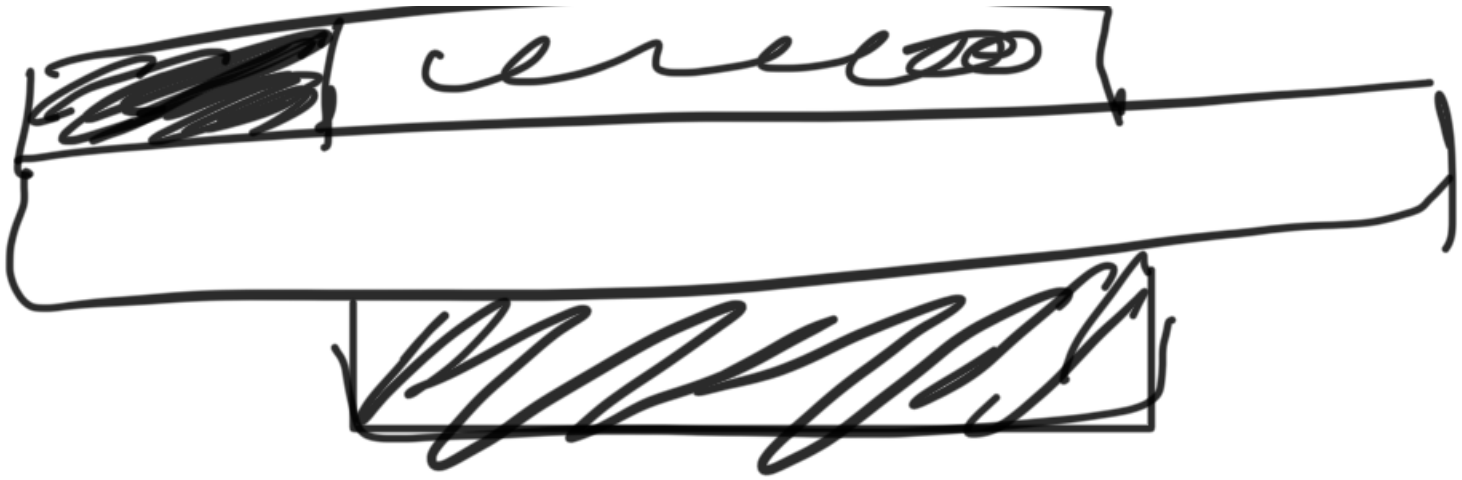
-1 max = -1

-7 max = -1

-1

How to find start and end of subarray.

Find many subarray sum which is less then given integer K.



arr = [-1, 3, -2, 4, 1, -3, 5]

preSum = [-1, 2, 0, 4, 5, 2, 7]

i j

max (preSum[j] - preSum[i]) < K

8

x = ~~4~~ 5

5

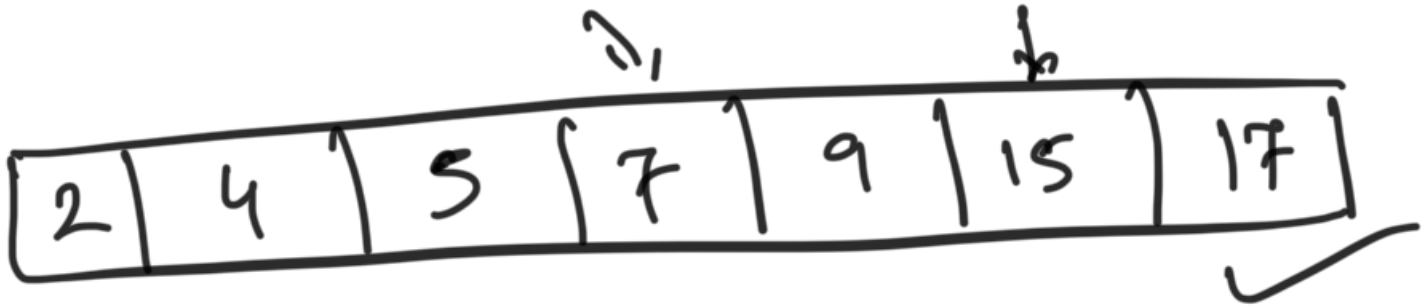
preSum[i] < 5

$p_j - p_i < K$

$p_i > K - p_j$

4

Presum[i] > 3



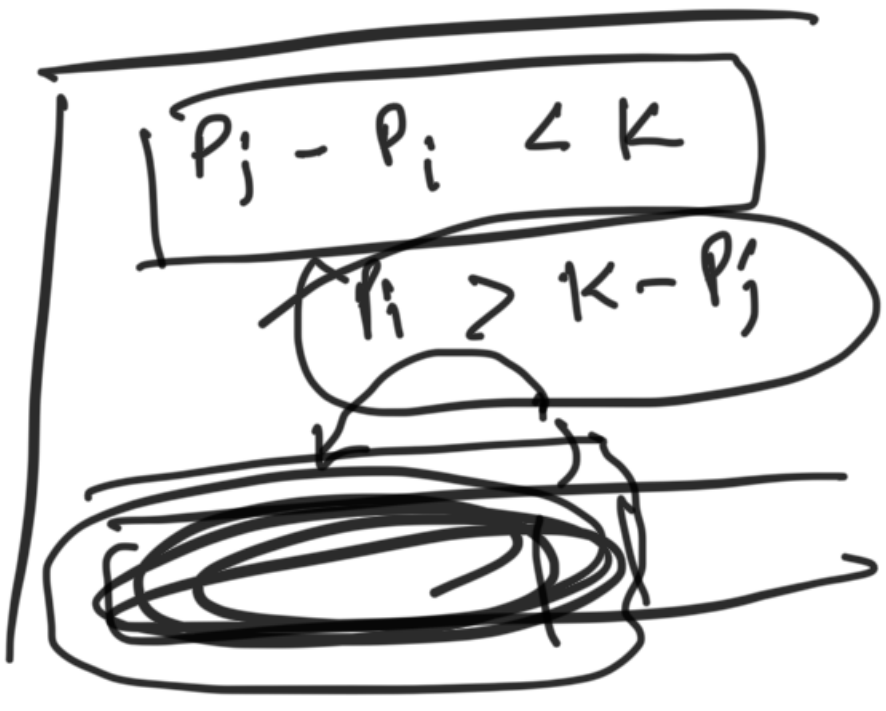
$x = 6 \rightarrow$

$x = 10$

maintain treeset

$pf = 0$

for x in arr



$ts.ceil(\frac{cs-k}{k})$:

$maxSum = max(cs - k, maxSum)$

$ts.add(cs)$

$[-4, 2, -1, 3, 8]$ $k = 5$

$p_j - p_i \leq 5$

$p_i > p_j - 5$

-4 -9 $TS = \{\ \}$

$TS = \{-4\}$

-2 -7

$p_i = -4$ $p_j = -2$

$-2 - (-4) = 2$ ✓

$TS = \{-4, -2\}$

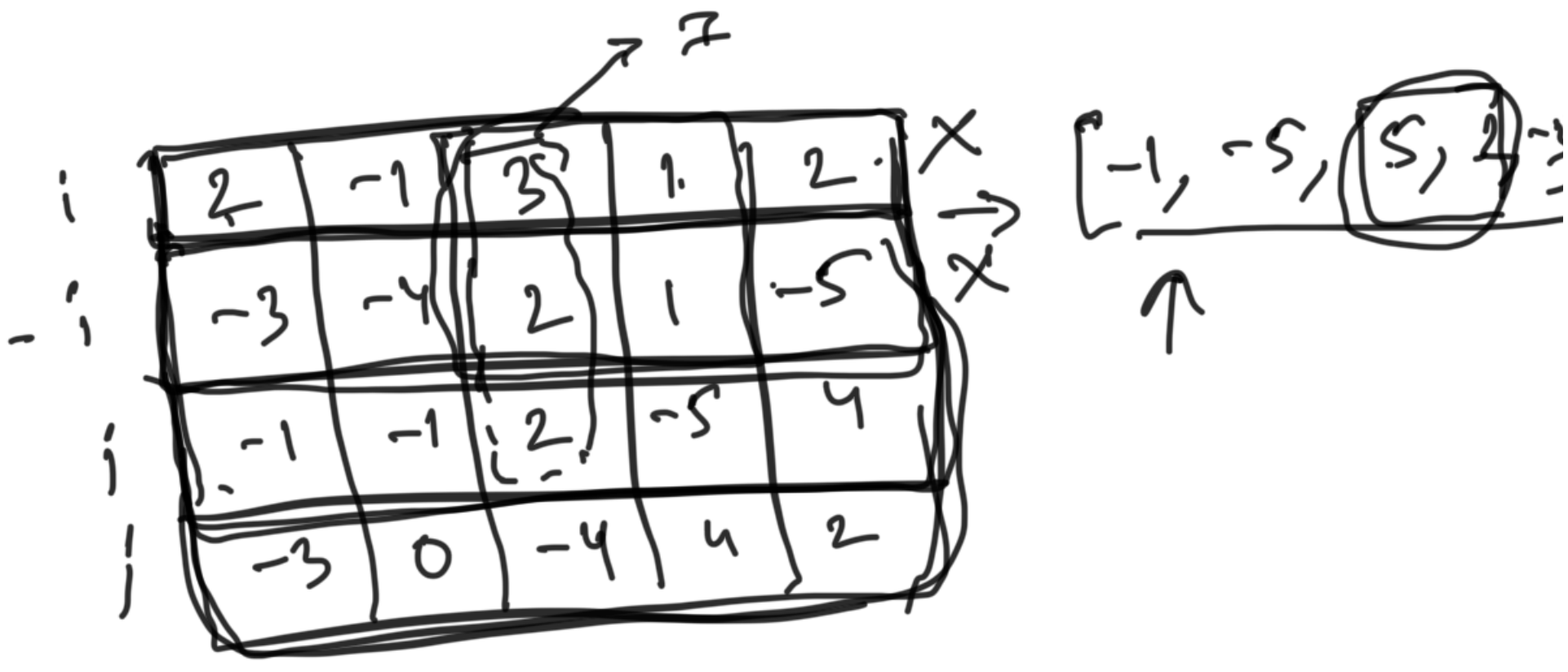
$-3 \qquad -8$

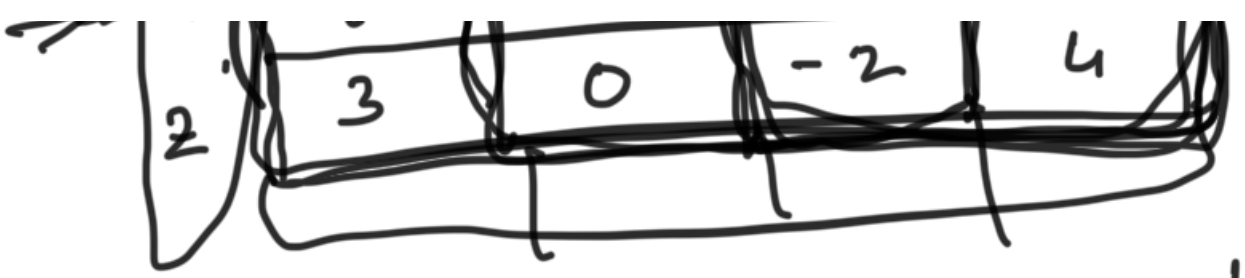
$-3, -(-4) = 1$ $TS = \{-4, -2, -3\}$

$0 \qquad -5$

$0 - (-4) = 4$ ✓✓ $TS = \{-1, -2, -3, 0$

$8 \qquad 3$


z
x
y
2 -1 3 1 2
-3 -4 2 1 -5
-1 -1 2 -5 4
-3 0 -4 4 2
[-1, -5, 5, 2
0
3 -1 2 4
-3 5 1
17



0 [3, -1, 2, 4] →

[0,1] [4, -4, 7, 5] ←

[0,1,2] [7, -4, 5, 9] —

1 [1, -3, 5, 1] —

[1,2] [4, -3, 3, 5] ✓

2 [2] [3, 0, -2, 4] —

Recursion

```
fun() {
    fun();
```

}

```
fun ($int n)
{   if( n<=0) return ).
    fun ( n-1 );
    Print n
}

main ( )
  {
     fun( 5 ):
  }
```

main → fun(5) → fun(4) → fun(3) → fun(2)

→ fun(1) → fun(0)

1
2
3
4
5

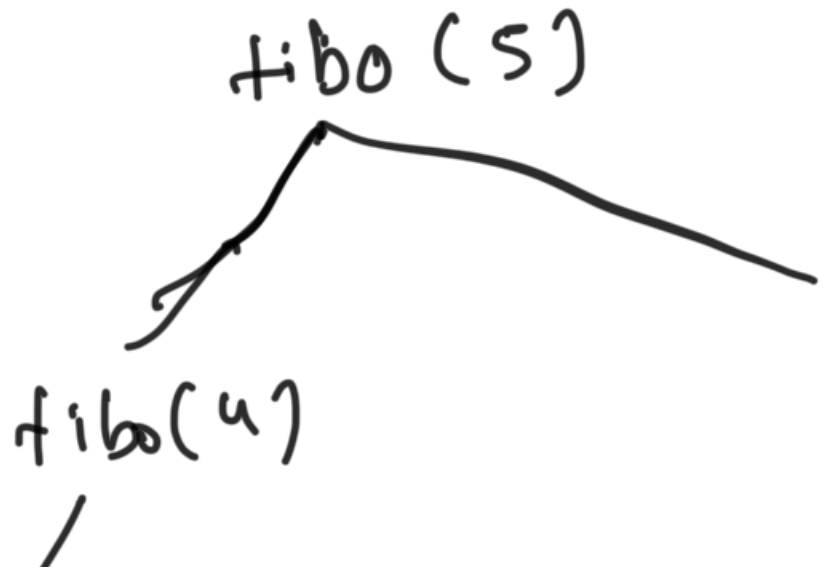
tibo — 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

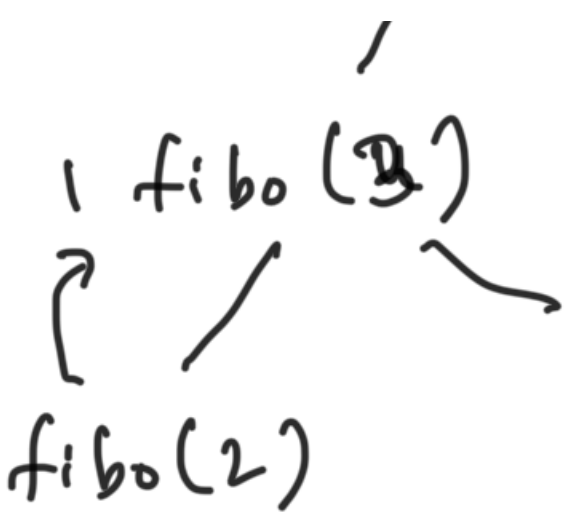
Daisy → 13, 21, 8

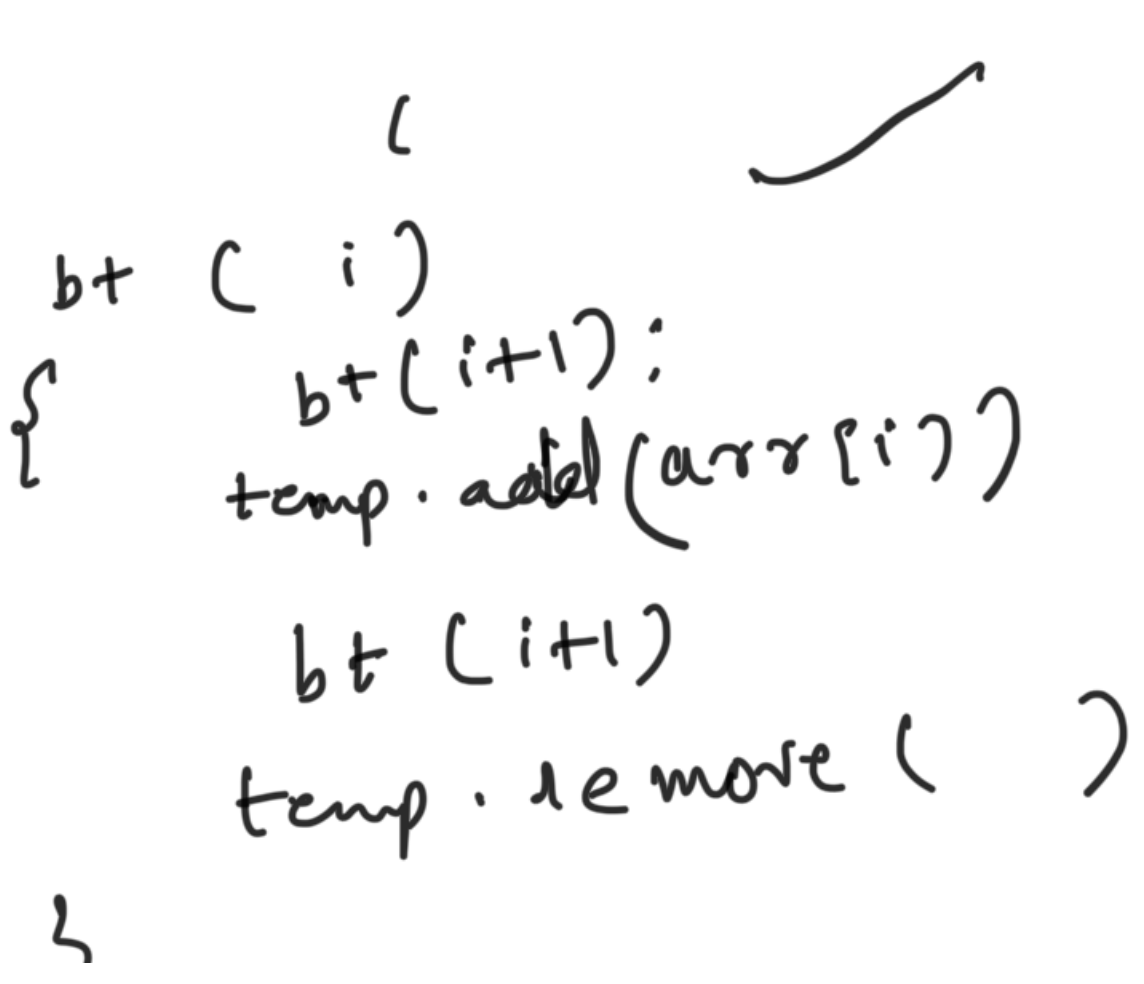
Sea waves → 3, 5, 8 — 1, 1, 2, 3,

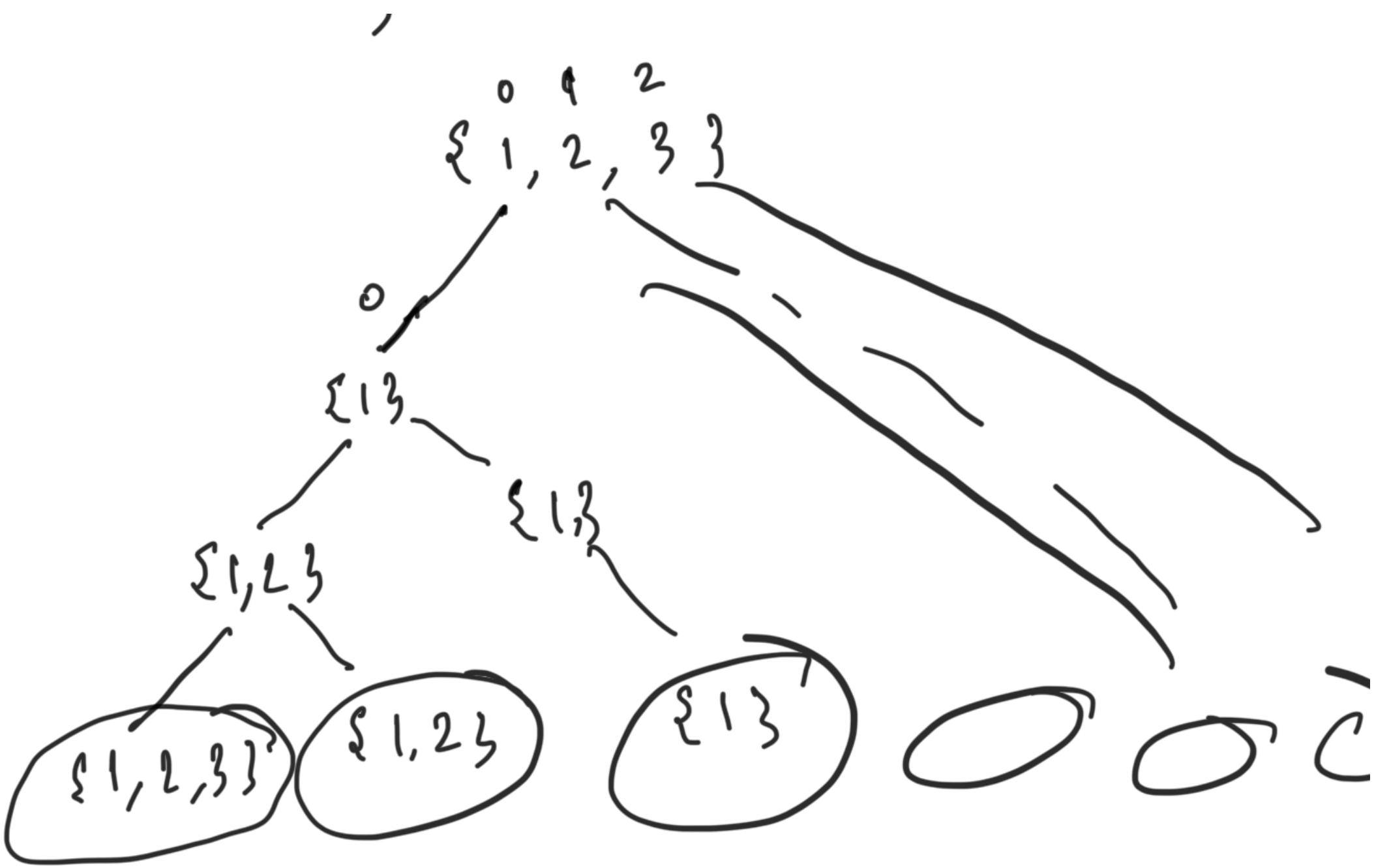
```
int fibo(int n)
{
    if(n==1) return 0
    if(n==2) return 1
    return fibo(n-1) + fibo(n-2)
}
```

n = 5




fibo(3)
fibo(2)


bt (i)
{
bt(i+1);
temp.add(arr[i])
bt (i+1)
temp.remove( )
}



Bit mask

{1, 2, 3}

0 0 0 0 — {}
1 0 0 1 — {3}

2	0 1 0	{ 2 }
3	0 1 1	{ 2, 3 }
4	1 0 0	{ 1 }
5	1 0 1	{ 1, 3 }
6	1 1 0	{ 1, 2 }
7	1 1 1	{ 1, 2, 3 }

n

from 0 to 2^n

for (i=0; i < 2^n ; i++)

{ (i)

}

}

add
call
remove

$[0 \qquad 2^5]$

0 2^{20}

$x << 1 = x \times 2$

$x >> 1 = x/2$

$1 << 20 = 2^{20}$

$$2^{20} = 1 << 20$$