Let's start at 8:32

Searching

→ Agenda → To learn basics of binary search.
→ To appreciate why binary search!
→ To check how we determine whether we can binary search.

→ Dictionary

Rain

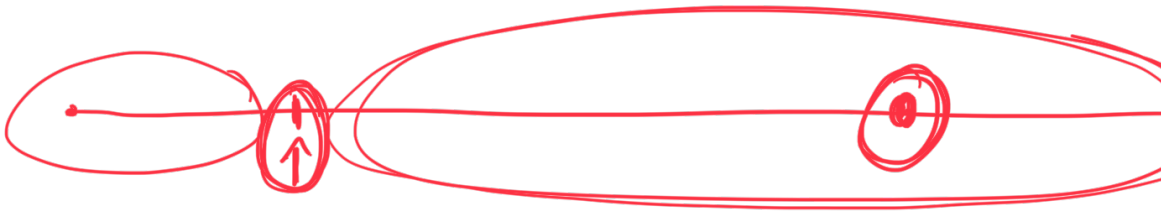
Ran

That's too slow

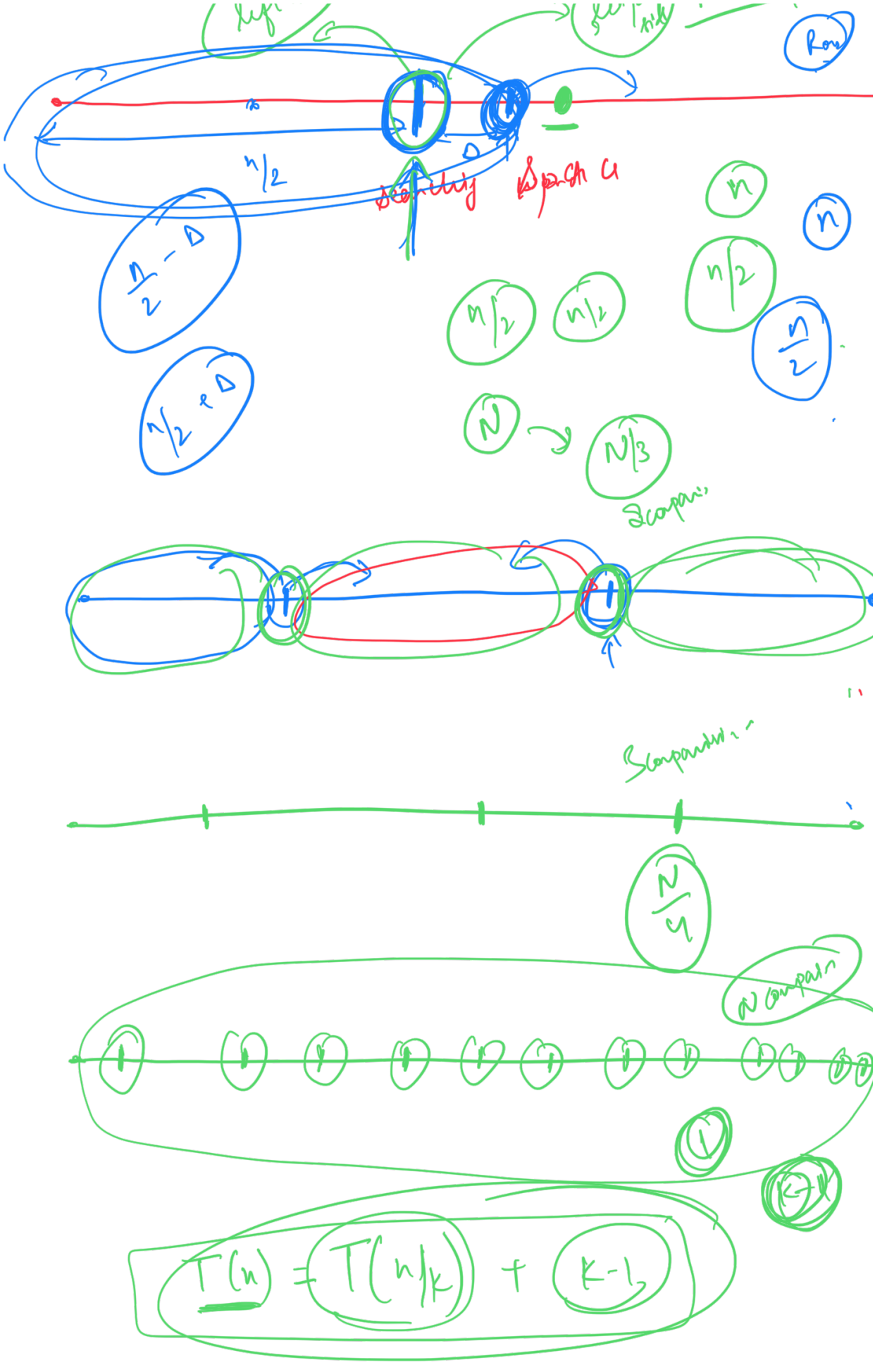
Can we make it fast?

Ran

→ When we are able to find some properties so that we can make searching fast.



Random point


n/2
n/2 - D
n/2 + D
n
n/2
n/2
n/2
n/2
N
N/3
N/4
T(n) = T(n/k) + k-1

$T(n) = T(n/2) + 1$

$T(n) = T(n/3) + 2$

$T(n) = T(n/2) + 1$

$= T(n/2^2) + 2$

$= T(n/2^3) + 3$

$= T(1) + \log_2 n$

$= \log_2 n \; T\left(\frac{n}{2^k}\right) + k$

$2^k = n$

$k = \log_2 n$

$\frac{n}{2^k} = 1$

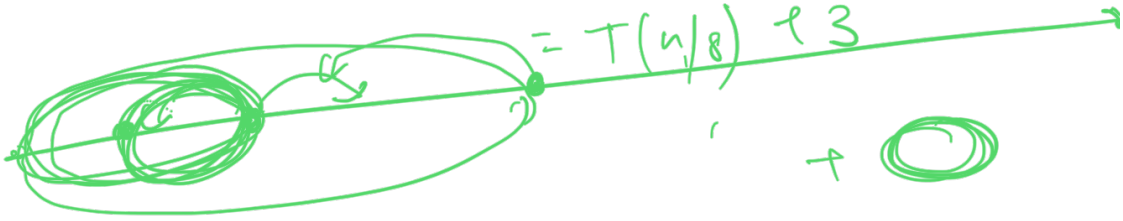
$n = 2^k$

$\log_2 n = k$

$T(n) = T(n/2) + 1$

$= T(n/4) + 2$

$= T(n/8) + 3$



$T(n) = T(1) + \log_2(N)$

$\log_2 N$

$T(n) = T(n/3) + 2$

$= 2\,T(n/3^2) + 2 \times 2$

$= T(n/3^3) + 2 \times 3$

$= 2\,T(n/3^k) + 2 \times k$

$2 \times \log_3 n$

$\frac{n}{3^k} = 1$

$n = 3^k$

$k = \log_3 n$

$T(n) = T(n/4) + 3 \times 1$

$= T(n/4^2) + 3 \times 2$

$\vdots$

$= T(n/4^k) + 3 \times k$

$\log_4 n$

$= 3 \log_4 n$

k random points

$K \log_{(K+1)} n$

$K \log_{(K+1)} n$

Time complexity

$K \geq 1$

value of K where we have min value of this function

$f(K) = K \log_{K+1} n$

$= K \dfrac{\log n}{\log K+1}$

$\log_a b = \dfrac{\log b}{\log a}$

$K \geq 1$

$f'(K) \geq 0$

$\dfrac{K}{\log K+1}$ $\quad K \geq 1$

$K \geq 1$

$\dfrac{\log(K+1) - \frac{K}{K+1}}{(\log(K+1))^2}$ $\quad K \geq 1$

$\log K+1 - \dfrac{K}{K+1}$

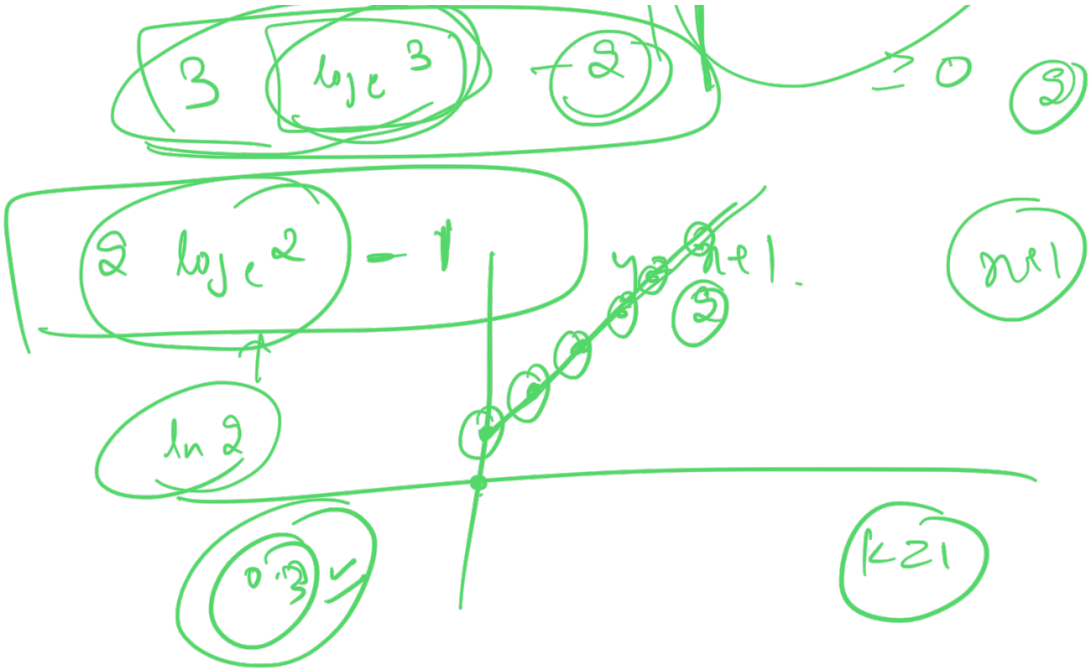
$(K+1)\log_e(K+1) - K$ $\quad \log_e(K+1)$

$2 \log 2 - 1$

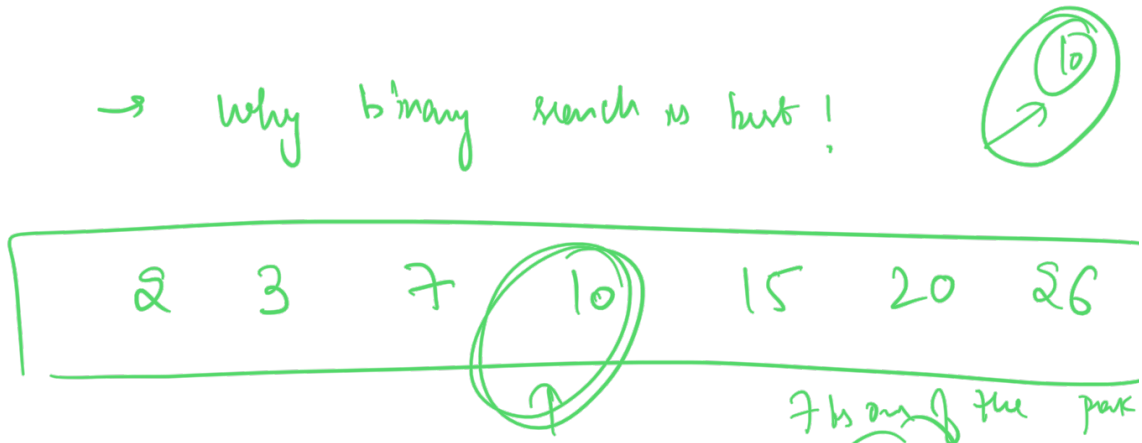
$\log$ $\quad 11 \log_{\frac{11}{10}} 12 - 10$

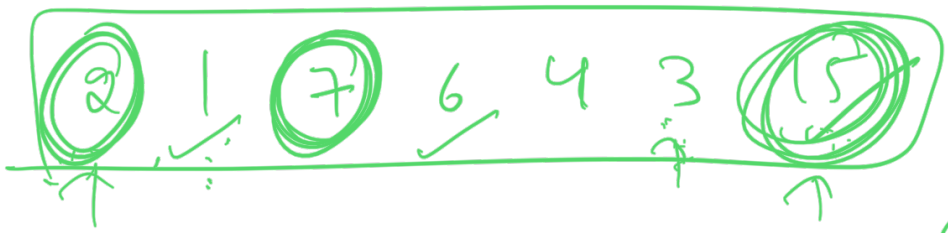
increasing fun

2



→ Why binary search is fast!





peak element in the array

You have been given an array and array is not sorted. You need to find peak elem in the array. And the peak element is defined as. i of $arr[i] \geq arr[i-1]$ and $arr[i] \geq arr[i+1]$ only if both of the neighbour

Max

O(n)

$T(n) = 2T(n/2) + 1$

$= 2^2 T(n/2^2)$

O(n)

$= 2^k + T(n/2^k)$

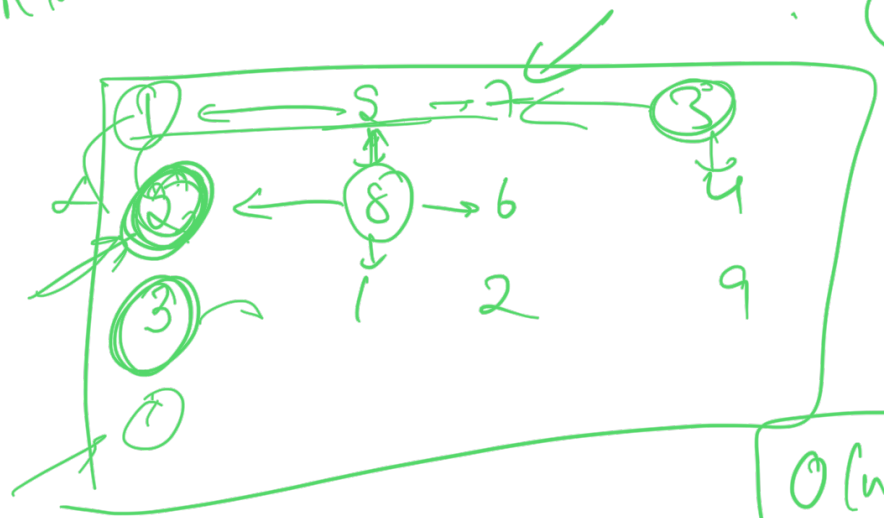
binary search

Recursion

n/2

n x m

5x4



O(n x m)

2D Array

Jagged

$x_4 < x_3$

$x_1 > x_3$

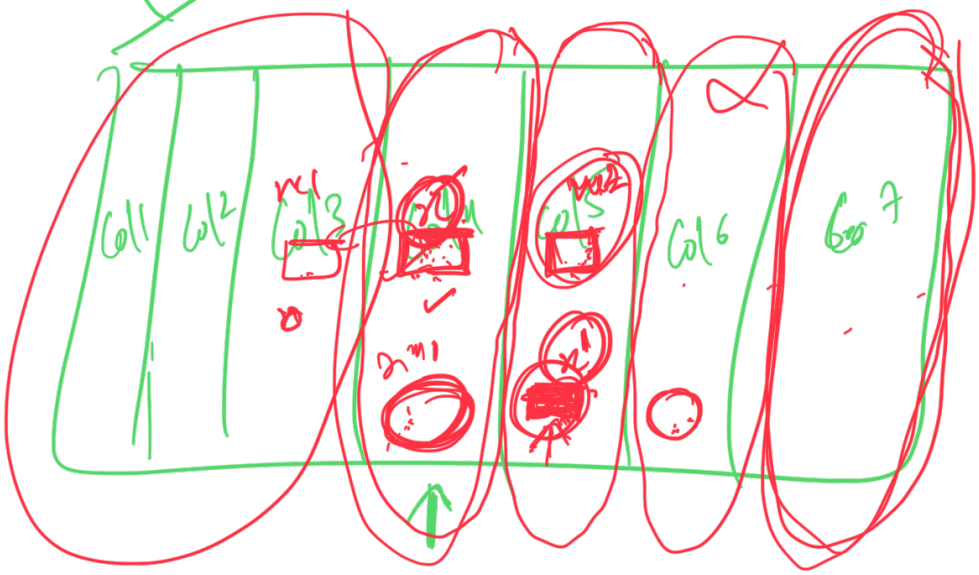
$x_2 > x_1$

$x_2 > x_3$

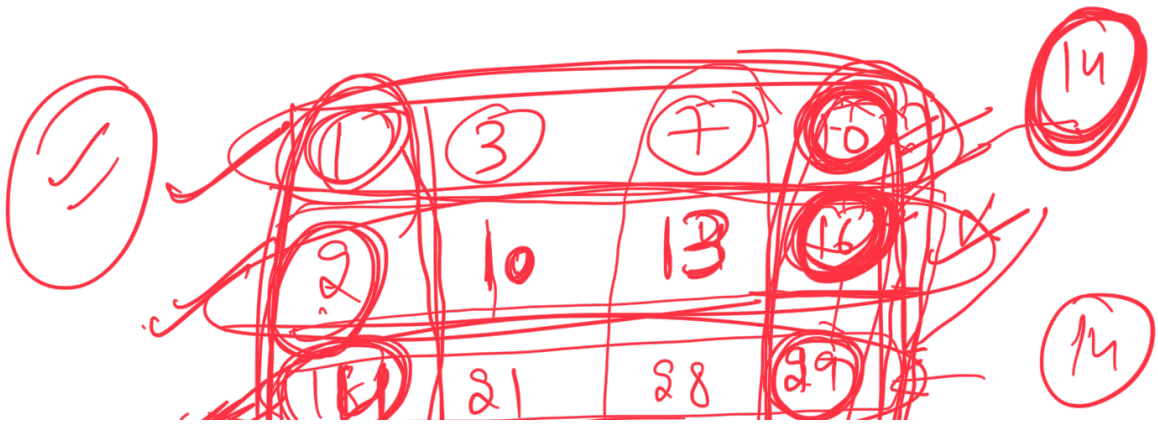
$x_2 > x_4$

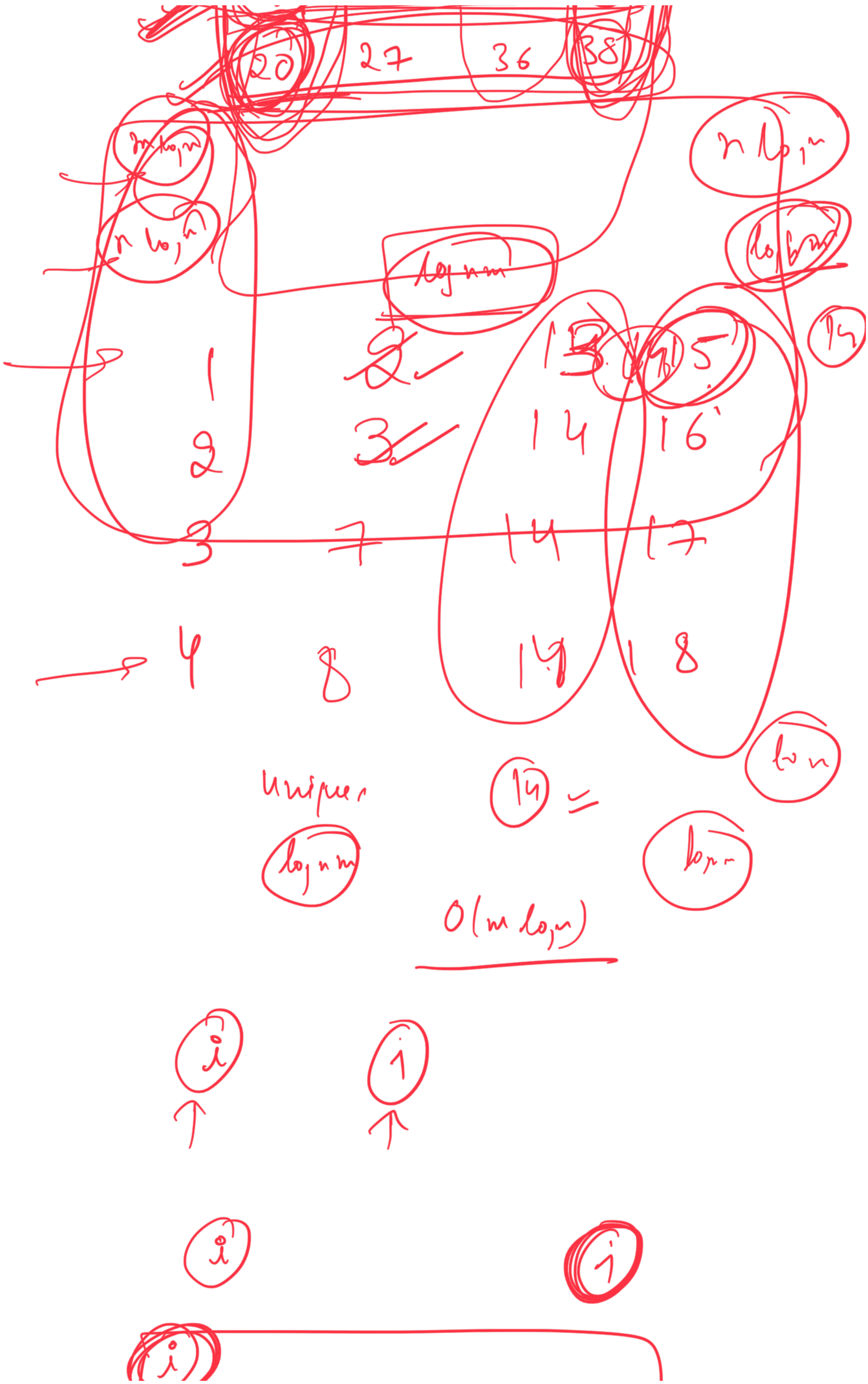
$x_1 > x_3$

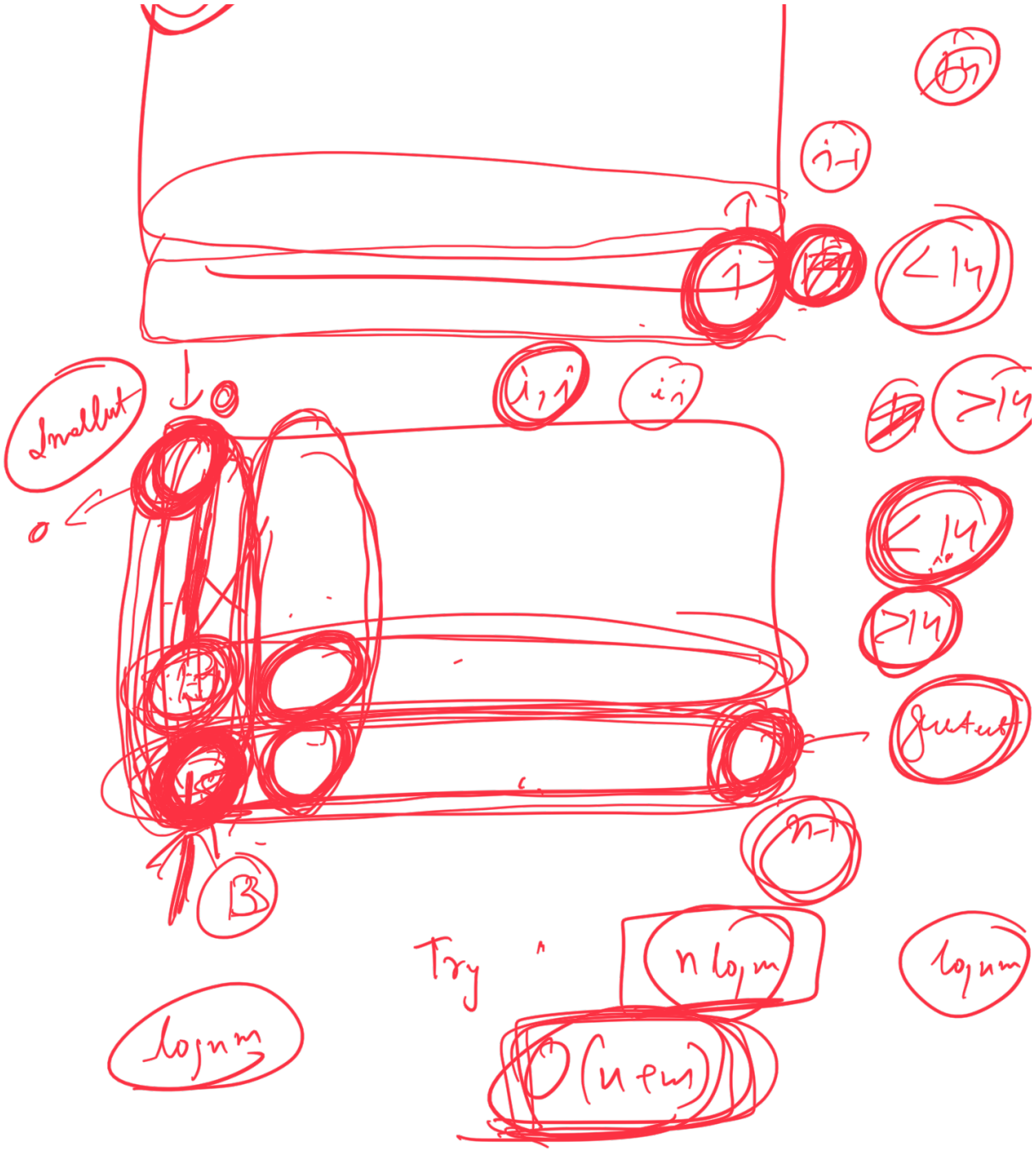
$x_2 > x_4$



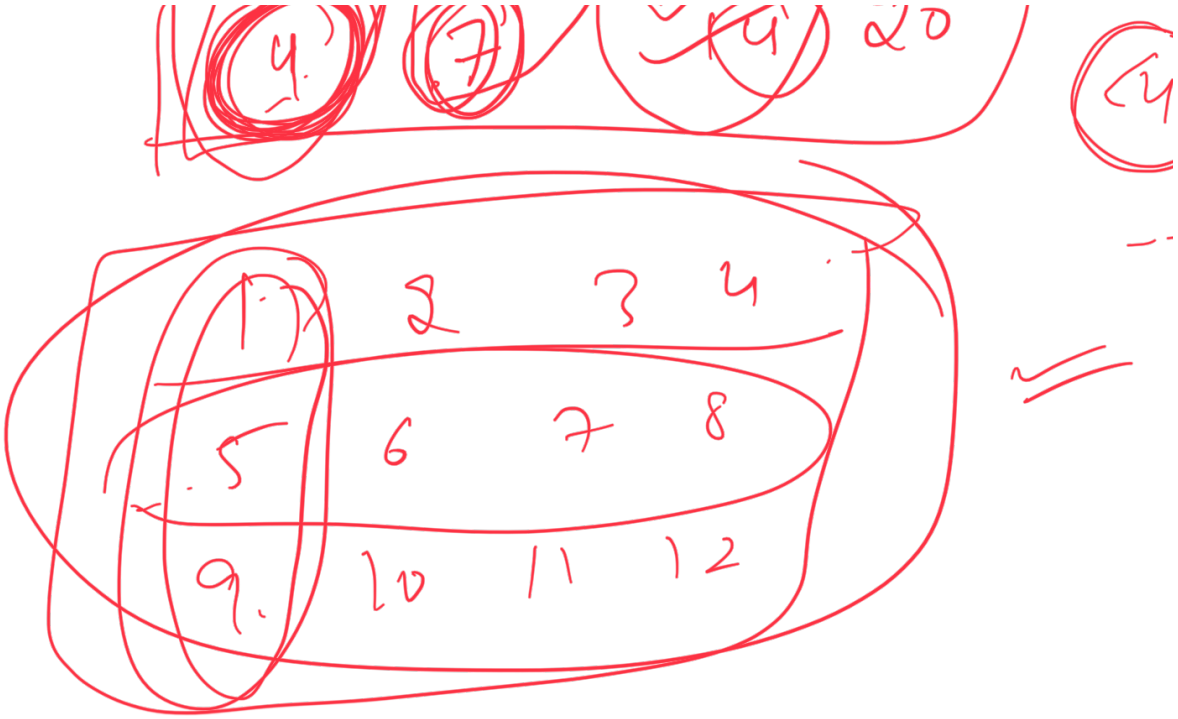
$x < m_2$




20 27 36 38
1 2 3 4
2 3 7 8
13 14 14 14
15 16 17 18
unique
O(m log m)


smallest
B
Try
n log m
log m
log m
(n+m)


log n
1
2
14
16
2
3
14
18
3
8
14
19


1
2
3
4
5
6
7
8
9
10
11
12