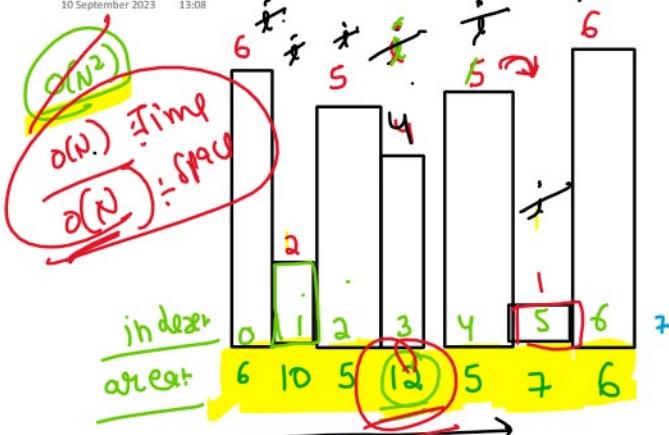


Largest rectangle in histogram

10 September 2023 13:08

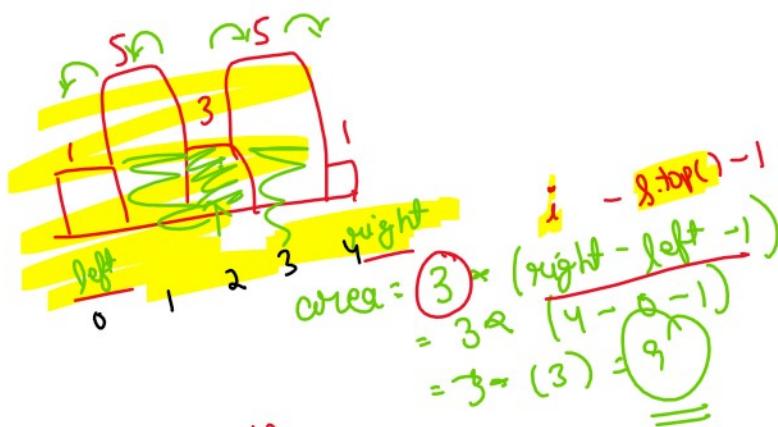


i array: [6, 2, 5, 4, 5, 1, 6]
width is 1 unit

$$\text{height} \propto (i - s.\text{top}() - 1)$$

$$\begin{aligned} 6 &= 1 = 6 \\ 5 &= (3 - 1 - 1) = 5 \\ 5 &= (5 - 3 - 1) = 5 \\ 4 &= (5 - 1 - 1) = 12 \\ 2 &\times 5 = 10 \\ 6 &= (7 - 5 - 1) = 6 \\ 1 &\times 7 = 7 \end{aligned}$$

- 1. If the new bar height is greater
push in stack.
- 2. Else pop the bar from stack and calculate the
area.

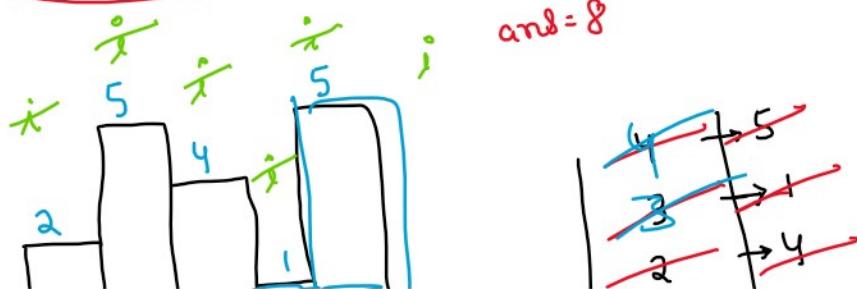


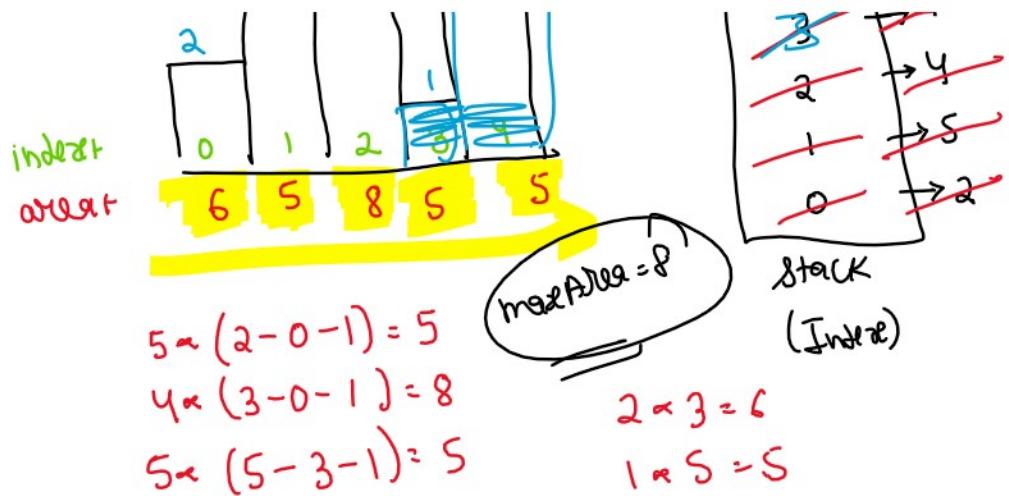
$$\text{Area} = \frac{\text{height} \times \text{width}}{i - s.\text{top}() - 1}$$

if stack is empty.

$$\text{height} \propto i$$

$$\text{height} \propto (i - s.\text{top}() - 1)$$





→ Maximal rectangle (DP)

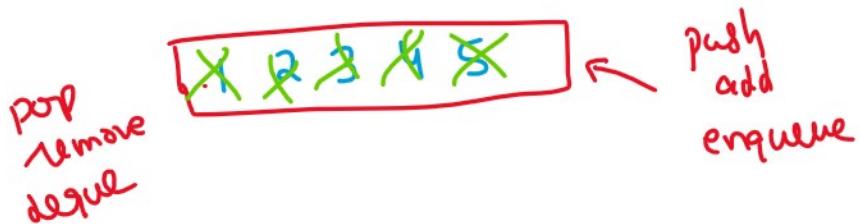
Queue

10 September 2023 13:56

FIFO
first in first out

Tree → level order traversal

Graph → BFS



{ atm queue

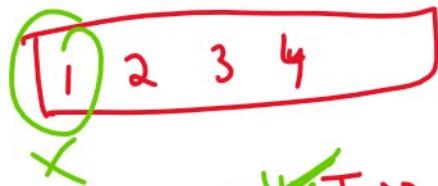
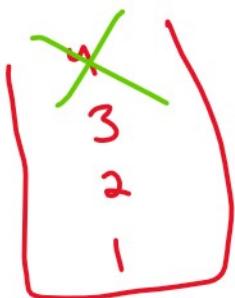
C++
push
pop
front
empty

Java
add / offer
remove / poll
peek
isEmpty



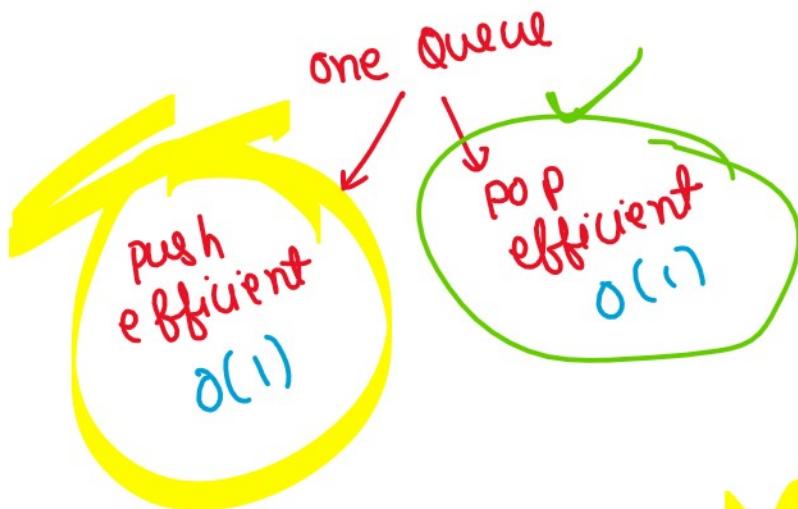
Implement Stack using Queue

10 September 2023 14:02



Two Queues

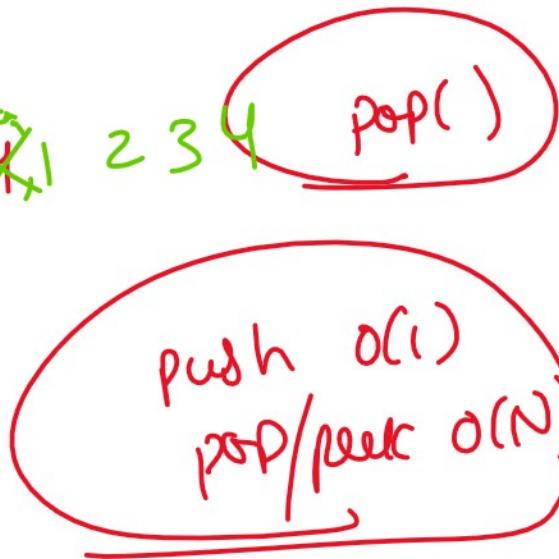
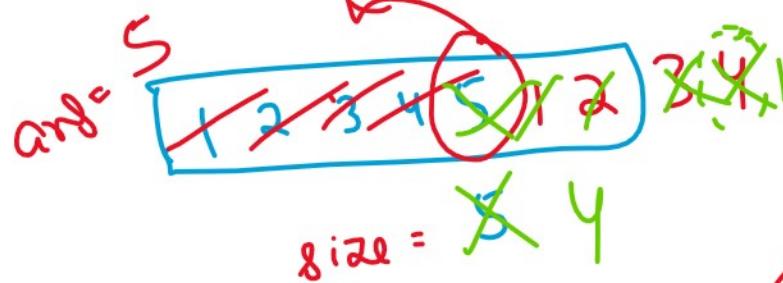
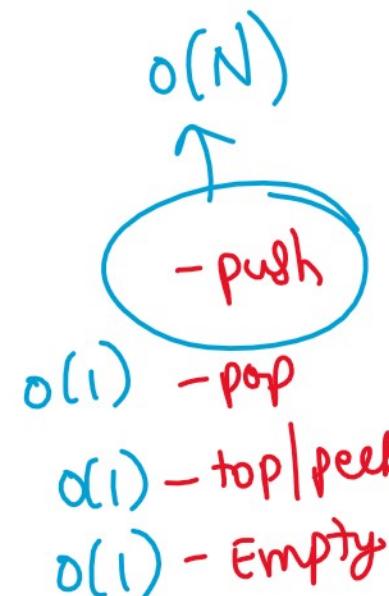
One Queue



~~size = 1~~

~~size = 2~~

size = 3



~~size = 5~~ 4

push $O(1)$

pop/peek $O(N)$

First non repeating char in a stream

10 September 2023 14:22

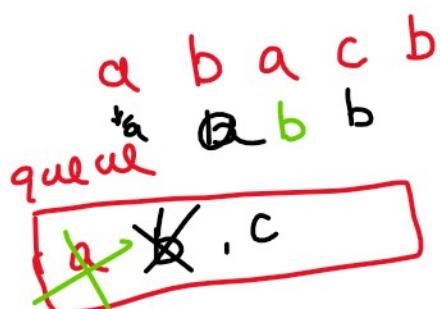
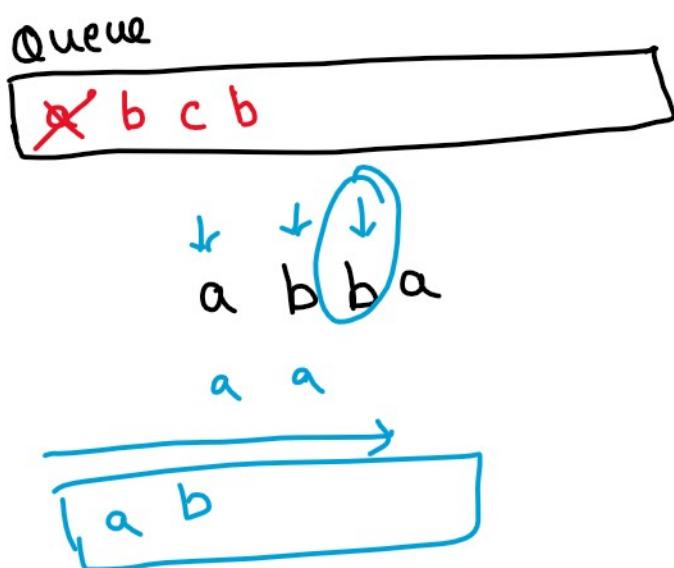
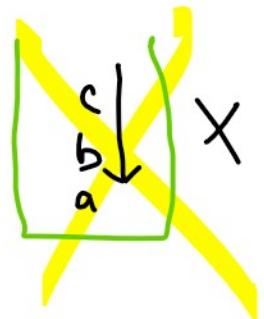
$$gtr = abacb$$

$$arr = aabbcc$$

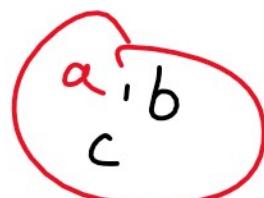
a b a c b
a a b b c

$$gtr = a a b c b a e c e d b f$$

$$a \# b b c c c d d \# \# f$$



set



$$gtr = \overbrace{a \bar{a} \bar{b} \bar{c} \bar{b} \bar{d} \bar{e} \bar{c} \bar{e} \bar{d} \bar{b} f}^{deced} \quad \text{underlined}$$

$$\underline{-\# b b c \quad c c d \quad d \# \# f}$$

input: u u
 \rightarrow a # b b c c c d d # # f

$O(1)$

map

a - 2

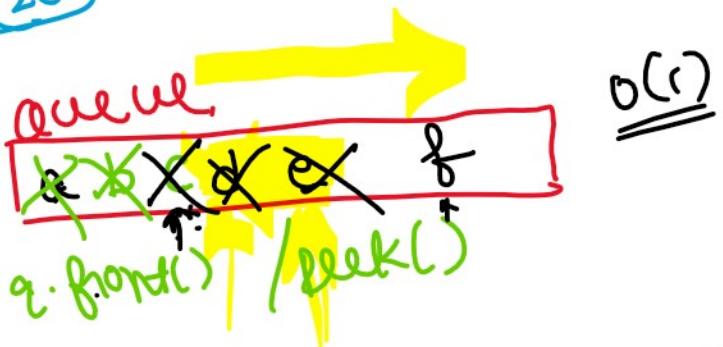
b - ~~x~~ 2

c - ~~x~~ 2

d - ~~x~~ 2

e - ~~x~~ 2

②6



$B-1$

Time $\leftarrow O(N)$
 Space $\leftarrow O(1)$

②6

Bit manipulation

10 September 2023 15:11

bitwise operators

&	AND
	OR
~	NOT
^	XOR

or gate

1 1 = 1
1 0 = 1
0 1 = 1
0 0 = 0

5 & 7

$$\begin{array}{r}
 5 = 0101 \\
 7 = 0111 \\
 \hline
 \text{And} \quad \underline{\underline{0101}} \quad (5)
 \end{array}$$

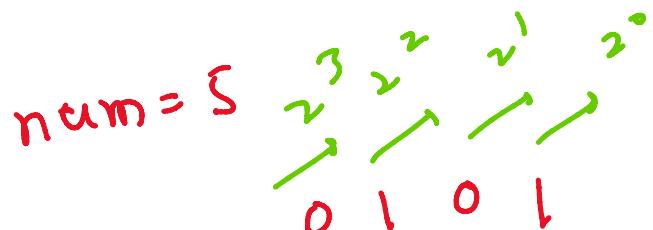
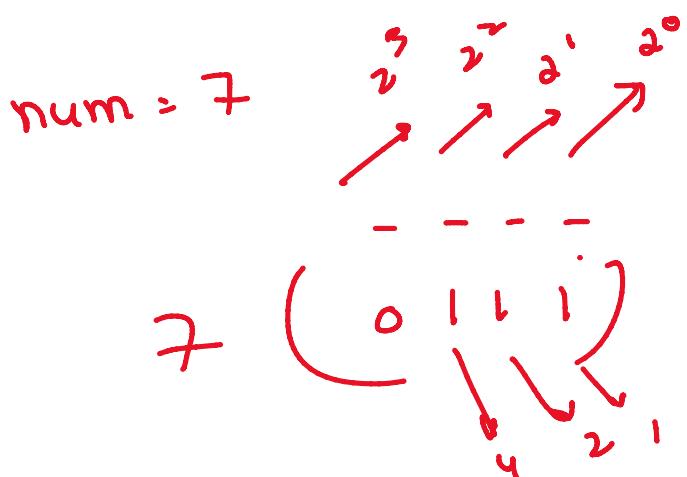
$$\cancel{\underline{\underline{5|6}}}$$

$$<= 0101$$

AND GATE

1 & 1 = 1
1 & 0 = 0
0 & 1 = 0
0 & 0 = 0

num = 7



$$\begin{array}{r}
 \text{5} \& \text{6} \\
 \hline
 \hline
 \end{array}$$

$$5 = 0101$$

$$\begin{array}{r}
 6 = 0110 \\
 \hline
 \text{And} \quad \underline{\underline{0100}} \quad (5)
 \end{array}$$

$$5 = 0101$$

$$6 = 0110$$

$$\begin{array}{r} \cancel{\text{OR}} \\ \hline 0111 \end{array}$$

4

XOR (\wedge) Same bits zero otherwise OR.

exactly 1 bit should be 1.

$$1 \wedge 1 = 0$$

$$1 \wedge 0 = 1$$

$$0 \wedge 1 = 1$$

$$0 \wedge 0 = 0$$

$$\underline{\underline{5 \wedge 7}}$$

$$5 = 0101$$

$$7 = 0111$$

$$\begin{array}{r} \wedge \\ 0010 \end{array}$$

3

$$5 = 0101$$

$$5 = 0101$$

$$\begin{array}{r} \wedge \\ 0000 \end{array}$$

$$\cancel{5 \wedge 3 \wedge 2 \wedge 5 \wedge 2}$$

$$= 3 \text{ ans}$$

XOR of same values is zero.

Shift operators

10 September 2023 15:19

1. Left shift ($<<$)
2. Right shift ($>>$)

↳ Left shift.

$$\begin{array}{r} 5 = 0 \ 1 \ 0 \ 1 \ 0 \\ 10 = 1 \ 0 \ 1 \ 0 \ 0 \end{array} \Rightarrow \begin{array}{l} 10 \\ 20 \end{array}$$

5
↓
10
↓
20

Multiply by 2.

$$5 <\!\!< 1 = 10$$

$$5 <\!\!< 2 = 20$$

$$5 <\!\!< 2^2 = 20$$

$$5 <\!\!< 3 = 5 <\!\!< 2^2 <\!\!< 2^1 = 40$$

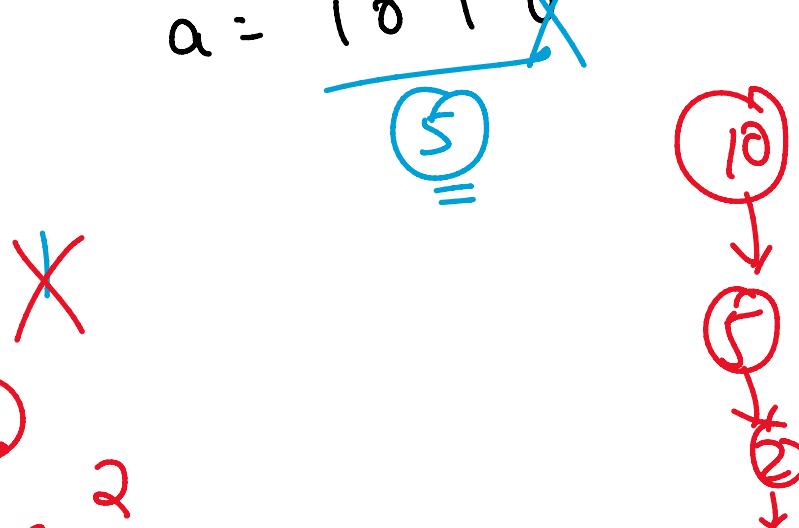
$$a <\!\!< b = a \cdot 2^b$$

$$a >\!\!> b = a / 2^b$$

2. Right shift

Divide by 2.

$$a = 1 \ 0 \ 1 \ 0$$



$$5 = 10 \times$$

$$\frac{1}{2}$$

$$n >\!\!> a = 10 = \frac{10}{2} = 2$$

$$10 \gg 2 = \frac{10}{2^2} = \frac{10}{4} = 2$$

$$10 \gg 3 = \frac{10}{2^3} = \frac{10}{8} = 1$$



Basic operations

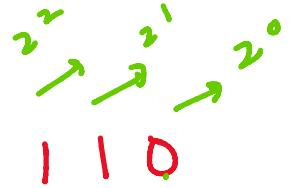
10 September 2023 15:25

odd or even

- 6 (even)

= 5 (odd)

= 10 (even)



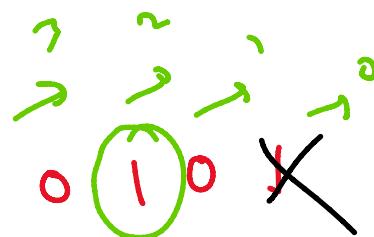
1 0 1

1 0 1 0

Get ith bit

n = 5

i = 2



ans = 1

0 1 0
0 1 1
—
0 1

(i)
1. 2 times right shift and take & with 1

n = 5

i = 2

2. mask

$1 \ll i$

$1 \ll 2$

(create mask by $1 \ll i$ and take &.)

8 0 1 0 1
 0 1 0 0
—
0 1 0 0



... ith bit

Set i^{th} bit

$$n = 5$$

$$i = 1$$

$$\text{mask} = \begin{cases} 1 & \text{at } i \\ 0 & \text{else} \end{cases}$$

$$\begin{array}{r}
 n \rightarrow 0 \ 1 \ 0 \ 1 \\
 \text{mask} \rightarrow 0 \ 0 \ 1 \ 0 \\
 \hline
 \text{ans} = 0 \ 1 \ 1 \ 1
 \end{array}$$

OR

3. Clear bit

$$n = 5$$

$$i = 2$$

$$\text{ans} = \underline{\underline{0 \ 0 \ 0 \ 1}}$$

$$\text{mask} = \begin{cases} 1 & \text{at } i \\ 0 & \text{else} \end{cases}$$

- 1. Create mask
- 2. Take negation
- 3. Take AND

$$\begin{array}{r}
 n = 0 \ 1 \ 0 \ 1 \\
 \text{mask} = \cancel{0 \ 1 \ 0 \ 0} \\
 \sim(\text{mask}) = \cancel{1 \ 0 \ 1 \ 1} \\
 \hline
 \text{ans} = \underline{\underline{0 \ 0 \ 0 \ 1}}
 \end{array}$$

Single number iii

10 September 2023 15:42

$$\text{arr} = \{ \cancel{4}, \cancel{2}, \cancel{7}, \cancel{4}, \cancel{1}, \cancel{4}, \cancel{3}, \cancel{3} \}$$

$$\cancel{7} \wedge 1 = \underline{\quad}$$

$$\text{ans} = [7, 1]$$

- ① Nested for loops. $O(N^2)$, $O(1)$
 - ② Sorting! $O(N \log N)$, $O(1)$
 - ③ Hashmap. Store frequency of number. $O(N)$, $O(N)$
 - ④ Bit. $O(N)$, $O(1)$
- Time ↑ Space ↑

$$\text{ans} = [3, 7]$$

$$\rightarrow \cancel{8} \cancel{1} \cancel{2} \cancel{3} \cancel{5} \cancel{7}$$

$$\text{res XOR} = \cancel{3} \wedge \cancel{7} = \underline{\quad}$$

$$\underline{\quad} \underline{\quad} = 4$$

$$\begin{array}{r} 0011 \\ 0111 \\ \hline 0100 \end{array}$$

a

$$\text{pos} = 2$$

$\cancel{101}$	$\cancel{001}$	$\cancel{010}$	$\cancel{001}$	$\cancel{010}$	$\cancel{011}$	$\cancel{101}$	$\cancel{111}$
5	1	2	1	2	3	5	7

$$\text{num1} = 5 \wedge 5 \wedge \cancel{7} = \cancel{7}$$

$$\text{num2} = \text{num1} \wedge \text{res} = \cancel{7} \wedge \cancel{4} = \cancel{3}$$

just 2

num2 = num1