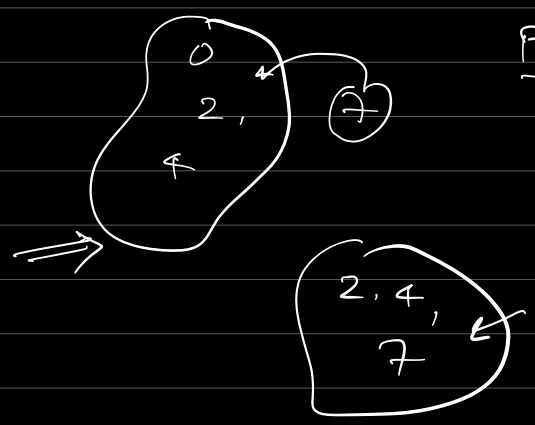
$$A = [\overset{0}{0}, \overset{1}{2}, \overset{2}{4}, \overset{3}{7}, \overset{4}{3}, \overset{5}{5}, \overset{6}{6}]$$

$K = 3$



$$P = 8 = \frac{1 \times 2 \times 4 \times 7}{1}$$

$$\text{New product} = \frac{P}{\text{element getting removed}} \times \text{new element.}$$

$\Rightarrow$ zero

$$\frac{P}{0}$$

Heap Sort

$\Rightarrow$ Using heaps to sort a given array.

$A \Rightarrow$ ascending order

$A \Rightarrow 10, 3, 5, 8, 1 \Rightarrow$ Min Heap

Sorted Array ⇒ 1, 3, 5, 8, 10

⇒ Convert the given input to a min Heap

⇒ Till the size of the heap becomes 0, delete min & add in sorted array.

$$T.C. = O(N) + O(\log N \times N)$$

$N = 5$

$$= O(N \log N) \times$$

10, ~~8~~, ~~5~~,
~~8~~, ~~1~~

5

$$\log(5) + \log(4) + \log(3) + \log(2) + \log(1)$$

$$\log(a) + \log(b) + \log(c) = \log(a \times b \times c)$$

$$= \log(5 \times 4 \times 3 \times 2 \times 1)$$

$$= \log(5!)$$

$$N \Rightarrow O(\log(N!))$$

$$N \Rightarrow \log(N) + \log(N-1) + \log(N-2) \ldots \log(1)$$

$$= \log(N \times N-1 \times N-2 \ldots 1)$$

$$= \log(N!)$$

Q What can you say about $\log(N!)$

1) $\log(N) + \log(N) + \log(N) \ldots\ldots$ N times $\rightarrow N\log(N)$

2) $\log(N) + \log(N-1) + \log(N-2) \ldots\ldots \log(1) \Rightarrow \log(N!)$

$$① > ②$$

$$\log(N!) \leq N\log(N)$$

Again

1) $\log(1) + \log(2) + \ldots \log\left(\frac{N}{2}\right) + \log\left(\frac{N}{2}+1\right) \ldots \log(N)$

2) $\underbrace{\log\left(\frac{N}{2}+1\right) + \log\left(\frac{N}{2}+2\right) + \log\left(\frac{N}{2}+3\right) \ldots \log(N)}_{N/2}$

$$① \geq ②$$

1, 2, 3, 4, 5, 6, 7, 8, 9, 10

6, 7, 8, 9, 10

2) $\Rightarrow \underbrace{\log\left(\frac{N}{2}\right) + \log\left(\frac{N}{2}\right) + \log\left(\frac{N}{2}\right) \cdots\cdots N/2 \text{ times}}$

$$\frac{N}{2}\log\left(\frac{N}{2}\right)$$

$$\log(N!) \geq \frac{N}{2}\log(N/2)$$

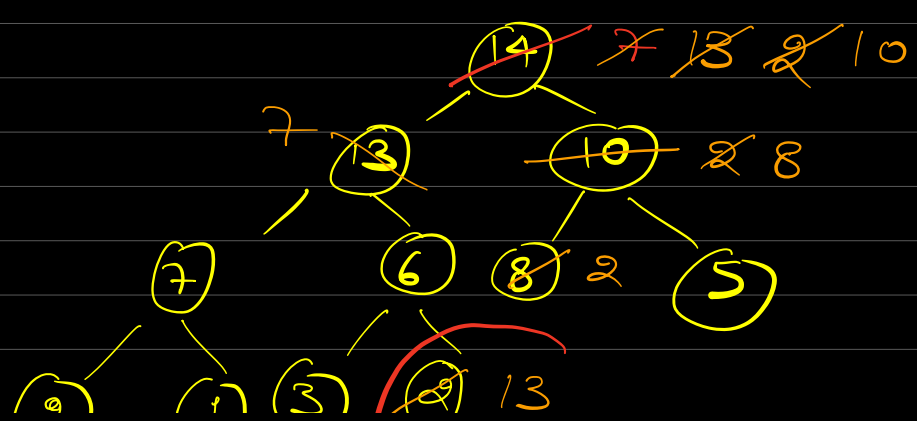
$$N\log(N) \geq \underline{\log(N!)} \geq \frac{N}{2}\log\left(\frac{N}{2}\right)$$

$\Downarrow$

Quicksort $\Rightarrow$ Randomized pivot

Space Complexity

$= O(N)$



$A \Rightarrow [10, 7, 8, 7, 6, 2, 5, 2, 1, 3, 13, 14]$

indices: 0 1 2 3 4 5 6 7 8 9 10 11

size

$SA \Rightarrow [\cancel{14},$

S.C. = O(1)

Ascending ⇒ Max Heap

Descending ⇒ Min Heap

Q Given an array with N elements s.t.

Every element is at max at a distance K from its position in the sorted array.

$|i - j|$

Sort the array

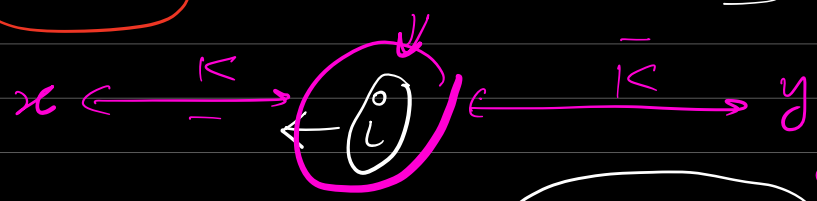
Eg

	0	1	2	3	4	5	6
A:	10,	18,	7,	5,	6,	22,	19
S:	5,	6,	7,	10,	18,	19,	22
	3	4	2	0	1	6	5

3	3	0	3	3	1	1

$K = 3$

$4 - 0 = 4 \times$

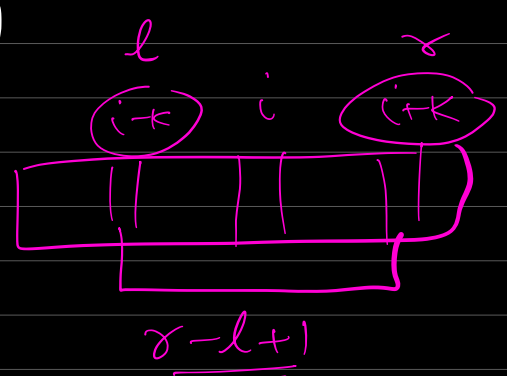


$i - x = K \Rightarrow x = i - K$ l

$y - i = K \Rightarrow y = K + i$ r

$$\begin{aligned} \#\text{ elements} &= y - x + 1 \\ &= (K+i) - (i-K) + 1 \\ &= K + i - i + K + 1 \\ &= 2K + 1 \end{aligned}$$

Heap of size $2K+1$



$x - 0 = K$

$x = K$

$K = 3$

$A \Rightarrow [10, 18, 7, 5, 6, 22, 19]$

0 1 2 3 4 5 6

7, 10, 18, 19, 22

$S \Rightarrow [5, 6, 7, 10, 18, 19, 22]$

0 1 2 3 4 5 6

K

[0, 3]

[0, 4]

$1-3 \leftarrow 1 \rightarrow 1+3$

$-3 \quad 0 \quad \rightarrow 4$

$\Rightarrow$ [18, 19, 22]

4

Min Heap

$\Rightarrow$ K + 1

[0, 4]

[0, 5]

[0, 6]

2K+1

T.C. = $O(2 \log(K+1) \times N)$

Adding $\Rightarrow$ Once $\log(K+1)$

Removing $\Rightarrow$ Once $\log(K+1)$

T.C. = $O(N \log K)$

Q K^th largest element of a given array.

T.C. = $K \log N$

Tries

Q Given an array of integers.

$A = [1, 2, 3, 5, 4]$

find a pair with max XOR

Sol^n ① Brute force ⇒ All possibilities

for every pair, calculate XOR & maintain a global max

T.C. = $O(n^2)$

② Sort & check for consecutive pairs.

T.C = $O(n \log n + N)$

③ Tries

A ⇒ [1, 2, 3, 5, 4]

bit

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

Dict ⇒ CAP LETTERS [A − Z]

26

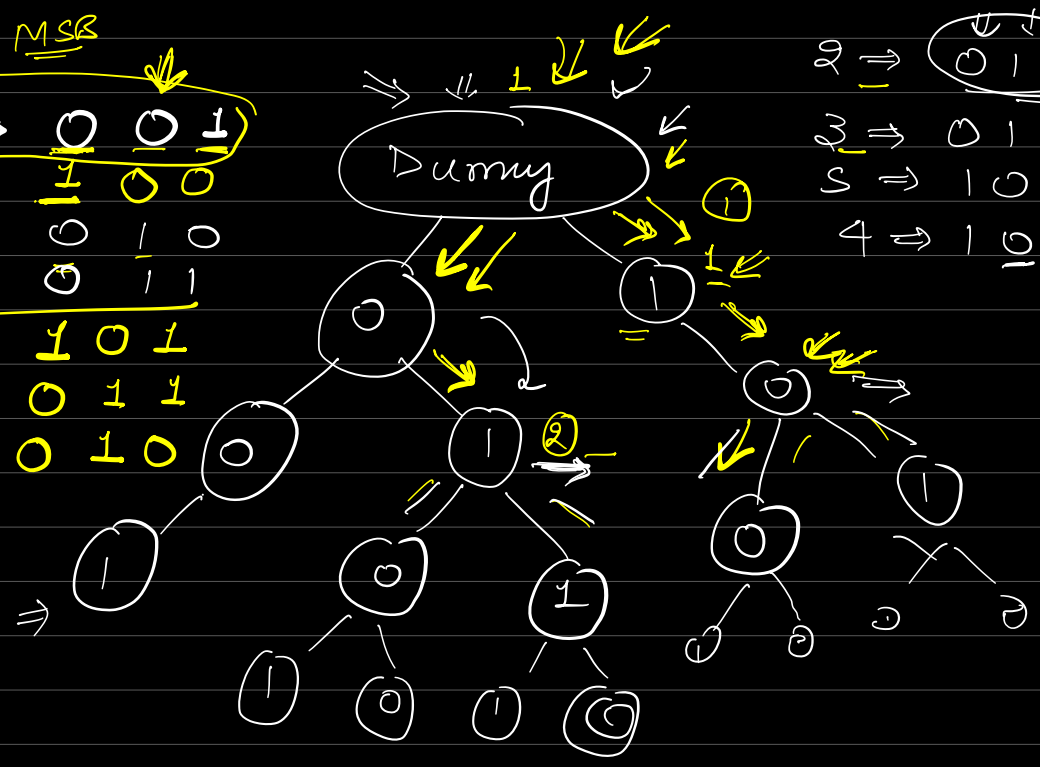
⇒ char ⇒ bit

32

MSB

1 ⇒	0	0	1
M ⇒	1	0	0
2 ⇒	0	1	0
3 ⇒	0	1	1
M ⇒	1	0	1
2 ⇒	0	1	1
3 ⇒	0	1	0

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



Inserting one int is $O(32)$

$= O(1)$

Time taken to create trie $\Rightarrow O(N)$

Q In how much time can you convert an int to binary.

Unsigned INT $\Rightarrow$ _ _ _ _ _ _ _ _

32

$N \,\&\, 1$ $\quad$ $N >> 1$

Code

```
for (i=0; i<32; i++){
bit = N & 1;
N = N >> 1;
}
```

T.C. $= O(32) = O(1)$

Intution

$1 \Rightarrow 0\ 0\ 1$ –

$M \Rightarrow 1\ 1\ 0$ – $= \underline{6}$

$1\ 1\ 1$

i^{th}

$1 \quad 1$

1	0	1	
2	1	0	
	$2^i = 1$	0	
$\Rightarrow$ 1 2	0	1	

$2^{i-1} + 2^{i-2} + 2^{i-3} \dots 2^0$

$2^i - 1$

Sol^n

$\Rightarrow$ 1) Create a bit trie $\Rightarrow O(N)$

2) $\Rightarrow$ for every integer $\Rightarrow O(N \times 32)$

$O(32)$ [$\hookrightarrow$ Try to search for the complement of every bit in the trie

max = ~~5~~ 7

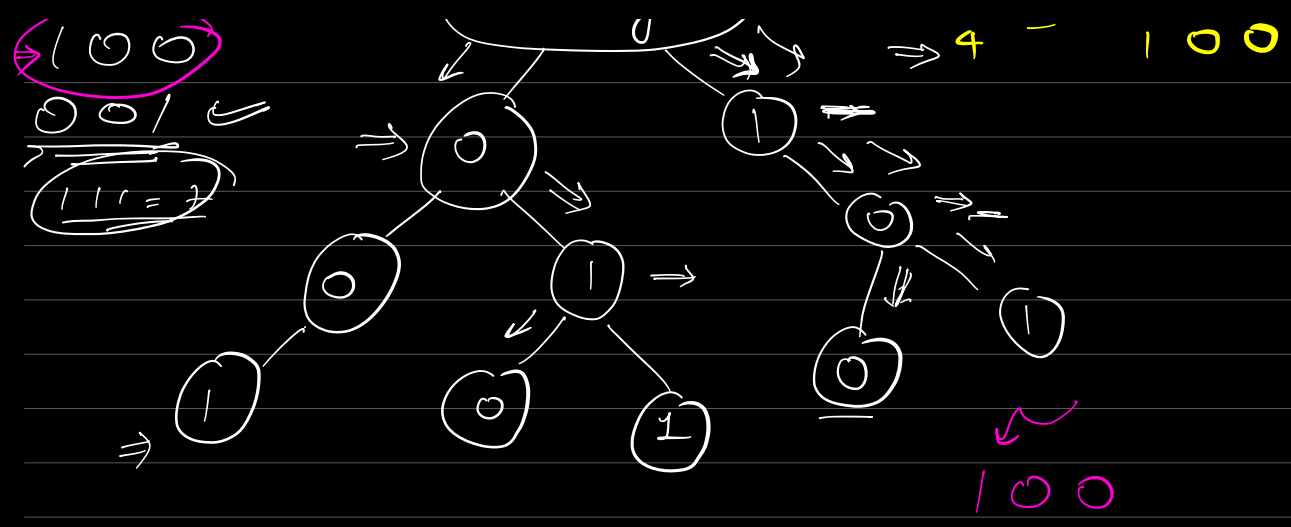
$\Rightarrow$ Dummy

$\Rightarrow 1$ – 0 0 1

$\Rightarrow 2$ – 0 1 0

$\Rightarrow 3$ – 0 1 1

$\Rightarrow 5$ – 1 0 1



T.C. = $O(N)$

S.C. = $O(N)$

Q Given an array of integer find max subarray XOR.

$A = [8, 1, 2, 12, 7, 6] \Rightarrow N$

$\#\text{subarray} = \frac{(n)(n+1)}{2} = O(n^2)$

Sol^n ① Brute force

$\forall$ subarray calculate the XOR

T.C. = $O(N^3)$

```
for (i = 0; i < N; i++) {
    for (j = i; j < N; j++) {
        for (k = i; k ≤ j; k++) {
            sum, XOR, max, min
```

~~Prefix~~

2 ⇒
1 ⇒

1 + 2 + 3 + 4 + 5 Carry forward

i
j j j j

Sum of a subarray for L to R

Prefix

$$XOR(A_i \text{ to } A_j) = \underbrace{XOR(A_0 \text{ to } A_j)}_{\text{Prefix XOR}[A_j]} \wedge \underbrace{XOR(A_0 \text{ to } A_{i-1})}_{\text{Prefix XOR}[A_{i-1}]}$$

⇒ A subarray with max XOR

⇒ i & j that gives me max XOR

Prefix XOR $[A_{i-1}]$ ^ Prefix XOR $[A_j]$

	0	1	2	3	4
		i		j	
A ⇒	1,	2,	3,	5,	4
		i-1		j	
P ⇒	1,	3,	0,	5,	1

⇒

0 0 1
0 1 0
0 1 1
1 0 1
1 0 0

1, 2, 3, 4, 5, 6

1 (2, 3, 4, 5, 6, 1 N-1

2 (3, 4, 5, 6, 1, 2 N-2

N-x

N

K > N

K % N = x

O(1)

(N − K % N)

O(N)

log(N)

MSB

	1	0	0	1	0	0	1
$M_1 \Rightarrow$	0	0	0	1	0	0	1
$M_2 \Rightarrow$	1	1	1	0	1	1	0
As $M_1 \Rightarrow$	1	0	0	0	0	0	0
$A_s \Rightarrow$	0	1	1	1	1	1	1

2^i $\quad$ $2^i - 1$