

\*\*\*  $1 \leq n$  represents  $2^n$  for a structure N of len(n). Num from

floor( $\log_{10}(\text{num})$ ) = no. of digits

## CONVERSIONS

0 to  $2^n - 1$  represents the complete subset for N (i.e. which element to include or exclude to get an element of the subset) (0  $\rightarrow$  exclusion, 1  $\rightarrow$  inclusion)

Binary to decimal

10010.01

$$\Rightarrow (2^3 \times 1) + (2^1 \times 0) + (2^0 \times 0) + \dots$$

negative powers for decimal places ones.

decimal to base b

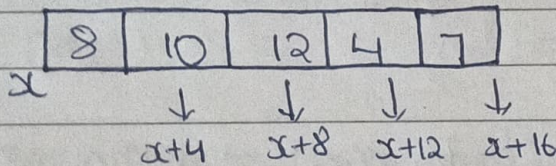
1056.24

$$\Rightarrow \begin{array}{c|c} b & 1056 \\ \hline b & x \end{array} \begin{array}{c} m \\ n \end{array} \cdot \begin{array}{c} 0.24 \times b \\ = p.ES \end{array}$$

(until termination)

## ARRAY MEMORY

int arr[5]; // 4 bytes in integer ::



for diff. data types byte size changes

eg: 

a	b	c
---	---	---

  
 $x \quad \downarrow \quad \downarrow$   
 $x+1 \quad x+2$

(as char  $\Rightarrow$  1 byte)

## BIT MANIPULATION & MATHS

1 byte = 8 bits

$a \ll 1 \Rightarrow 2^1 \times a$

$a \gg 1 \Rightarrow a \times 2^{-1}$

$a \& 1 = a$

$a \wedge 1 = \bar{a}$ ,  $a \wedge 0 = a$ ,  $a \wedge a = 0$

$N = a \mid b$ ,  $-N = \bar{a} \mid b$ , right most set bit isolation =  $N \& -N$

for n bits, range =  $-2^{n-1}$  to  $2^{n-1} - 1$

no. of digits in base b =  $(\text{int})(\log_b n) + 1$  (for num n)

1000 = 111 + 1

sieve of eratosthenes  $\Rightarrow$  if a is prime then an is also prime

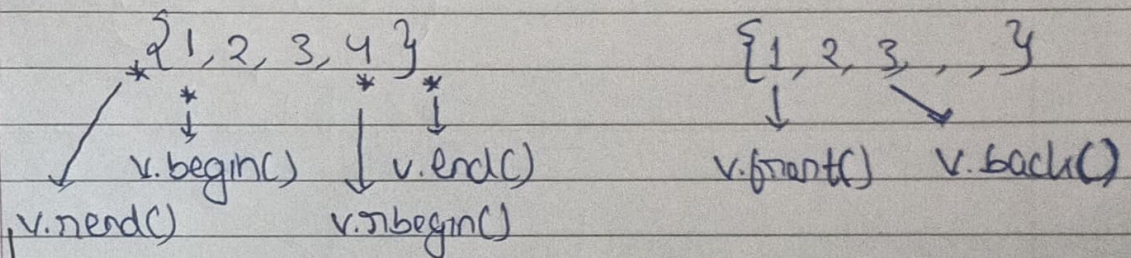
newton raphson method  $\Rightarrow \sqrt{n} = 0.5 \times (x + n/x)$ ;  $|\sqrt{n} - x| = \epsilon$

\*\*\*

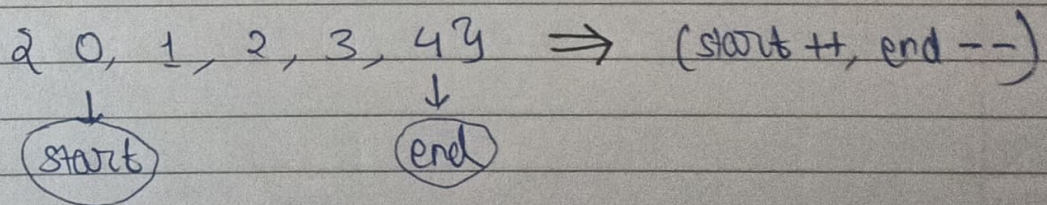


# VECTORS

- ★  $\{1, 2, 3\} \Rightarrow \text{size} = 3, \text{capacity} = 3$   
add 4  $\Rightarrow \{1, 2, 3, 4, \dots, \}$   $\Rightarrow \text{size} = 4, \text{capacity} = 6$
- ★  $O(1)$  time complexity for adding elements (total avg.)
- ★ get stored in heap memory (abundant) ~~not~~ unlike that of array (stack memory (limited))



## 2 POINTER



## PREFIX SUFFIX KADANE'S ALGO

$[1, 2, 5, -2, 4]$

prefix arr  $\Rightarrow [1, 3, 8, 6, 10]$       suffix arr  $\Rightarrow [13, 12, 10, 2, 4]$

Kadane's algo  $\Rightarrow$   $\text{max sum} = \max(\text{prefix}, \text{max sum})$   
if  $\text{prefix} < 0 \Rightarrow$  do  $\text{prefix} = 0$   
~~and~~ (start new subarray)

else continue extending current subarr. PAPER MINT



# 2D ARRAYS

	0	1	2
0	00	01	02
1	10	11	12
2	20	21	22
3	30	31	32

0	1	2	3	4	5	6	7	8	9	10	11
00	01	02	10	11	12	20	21	22	30	31	32

(system memory)

(CONTINUOUS  
MEMORY ALLOCATION)

★  $INDEX = \text{row index} \times \text{cols} + \text{col index}$

★  $\text{row index} = INDEX / \text{cols}$

★  $\text{col index} = INDEX \% \text{cols}$

★  $\text{address of arr}[i][j] = \text{base address} + INDEX \times \text{size of } \text{data type}$

★ eg  $\Rightarrow$  " "  $\text{arr}[2][1] =$  " "  $+ (2 \times 3 + 1) \times 4$