ARRAY

Primitives (int, float---) ⟺ Objects

Storage is diff

Memory Level

```
int [] arr = new int [size];
```

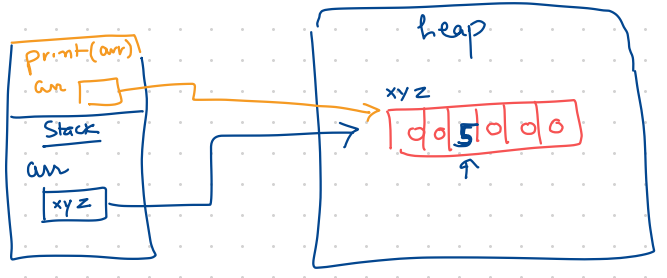
arr → Stack

new int [size] → heap

i=2 , data=5

```
arr[i] = data;
```

Pass By value



Object Scanner sc ; ← Single Object

class PaintBrush {

- color
- size
- width

}

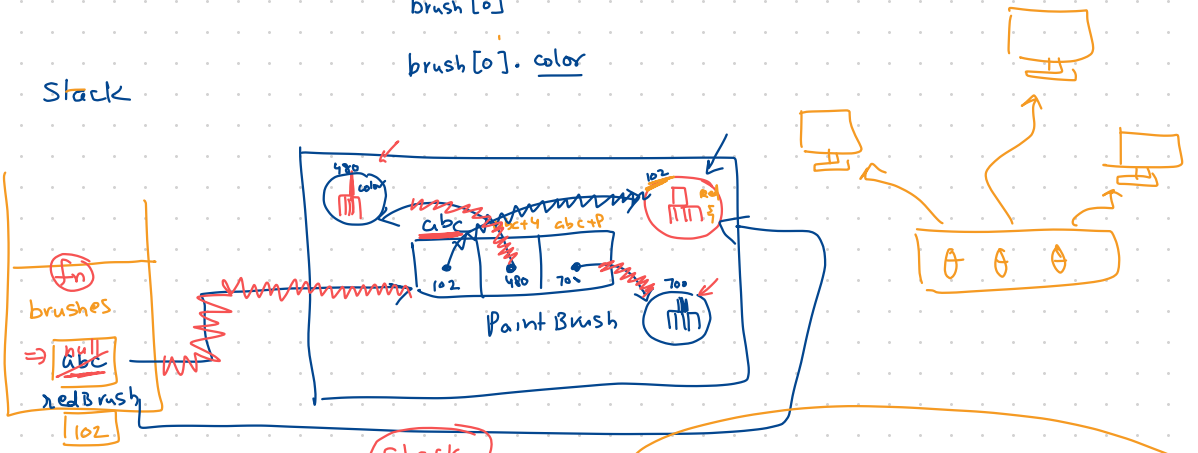
PaintBrush [ ] brushes = new PaintBrush [3] ; ⇐

Stack

brush [0]

brush [0]. color

Stack



brushes[0] = null ;

brushes = null ;

Stack

PaintBrush redBrush = new PaintBrush (Red , 5 , 2) ;

Array

brushes [0] = redBrush ;

brushes [1] = new PaintBrush ( ) ;

return address

Searching → Linear Search
Searching → Binary Search

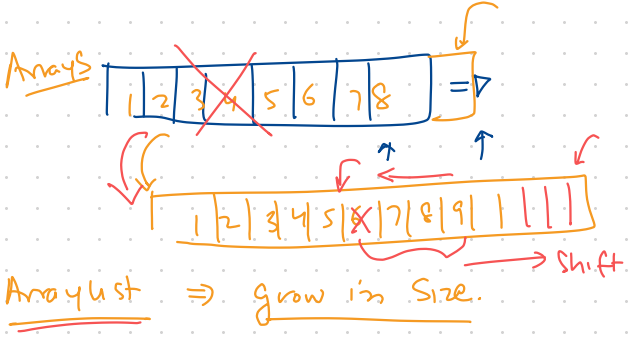
Phone / Whatsapp

Group, Ppl

Amazon

Zomato

Instagram

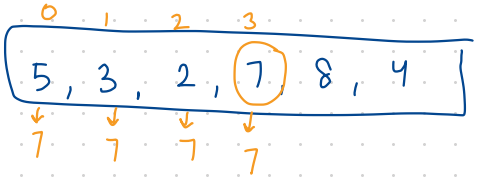


ArrayList ⇒ grow in Size.

~~Insert~~
~~Search~~
~~Deletion~~
~~Update~~

a[idx] = some-new-val;

Linear Search

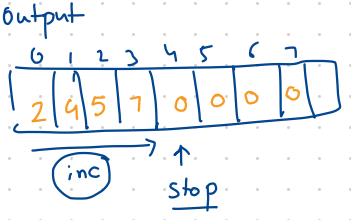


(I)

Note it down the indices

N + N
= 2N iterations.

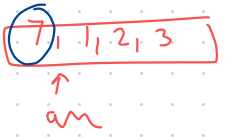
(II)

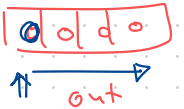


⇒ Interpret this array in main

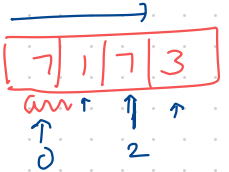
2, 4, 5, 7

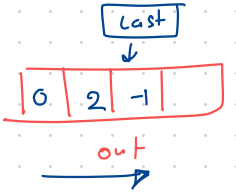
one case it can fail





Confusion?



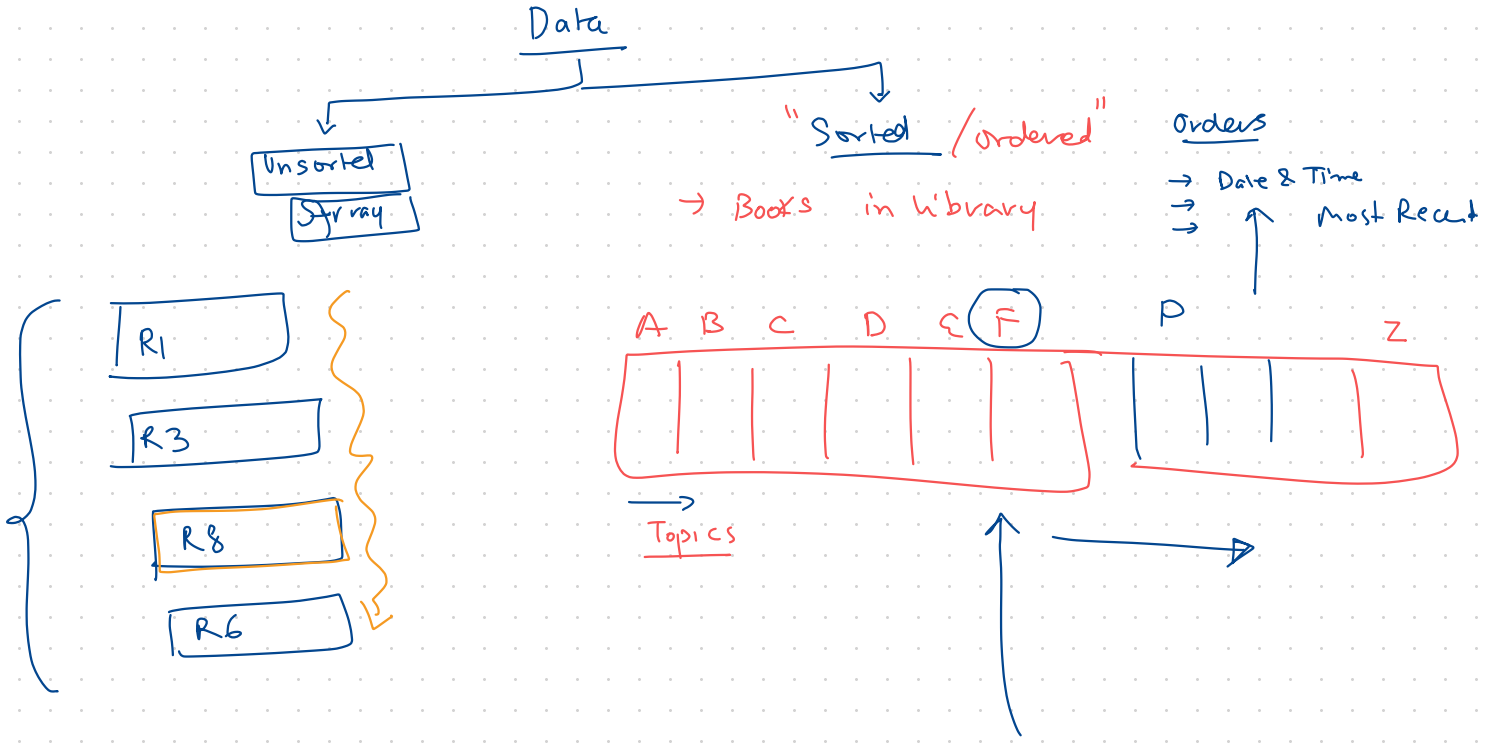


0, 2

Output

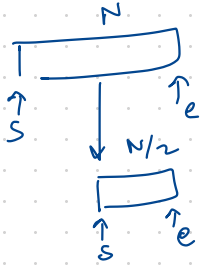
arr length + 1 ;

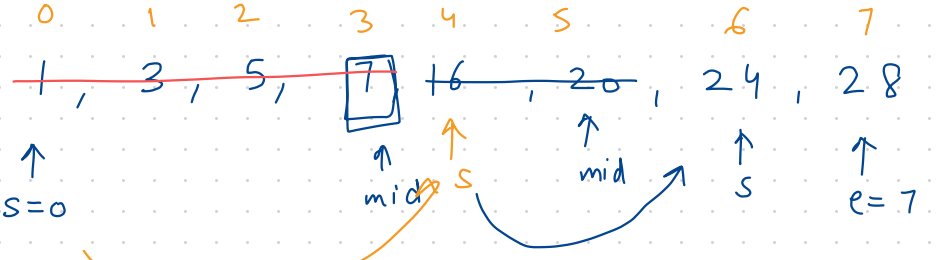
III Arraylist (grows automatically)


Data
Unsorted
Array
"Sorted / ordered"
→ Books in library
Orders
→ Date & Time
→
→
Most Recent
R1
R3
R8
R6
A B C D E F P Z
Topics

Binary Search

(only sorted arrays) inc / dec





Key = 24

✓ 0-7 $mid = \left(\frac{s+e}{2}\right) = \left(\frac{0+7}{2}\right) = (3)$

7 < 24

s = mid + 1 (Moved Right)

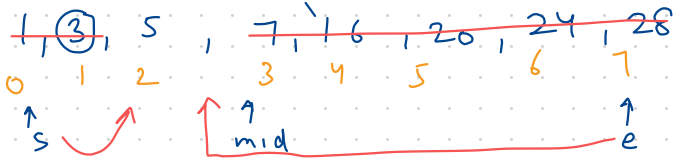
4-7 $mid = \left(\frac{s+e}{2}\right) = \left(\frac{4+7}{2}\right) = (5)$

20 < 24

s = mid + 1

8-7 $mid = \frac{6+7}{2} = (6)$

24 == 24



key = 5

0 - 7 mid = $\left(\frac{0+7}{2}\right)$ = 3

5 7

e = mid - 1

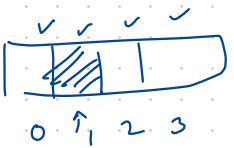
0 - 2 mid = $\left(\frac{0+2}{2}\right)$ = 1

3 5

s = mid + 1

2 - 2 mid = $\left(\frac{2+2}{2}\right)$ = 2

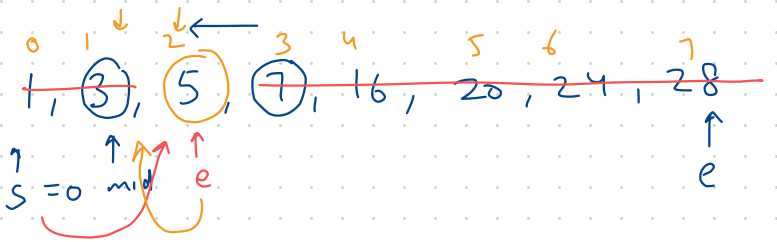
s == e



$\left(\frac{3}{2}\right)$ => 1

e s

Element not present



key = 4

0 - 7 mid = 3

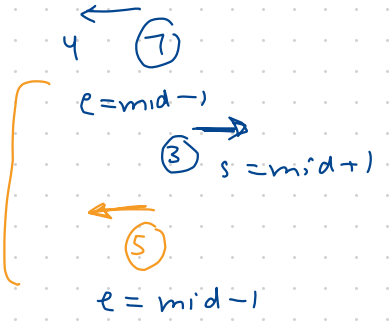
0 - 2 mid = 1

2 - 2 mid = 2

e = 1
s = 2

conclude?

NOT found

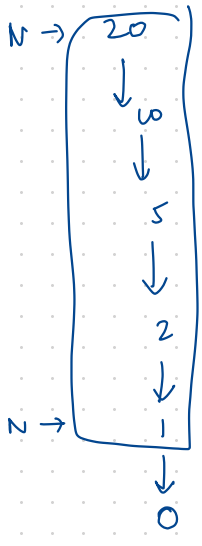


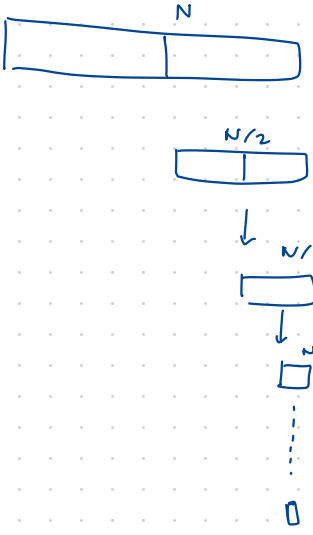
while (s <= e){

=

3

NOT found





Steps			
1	→	$N/2$	$= N/2^1$
2	→	$N/4$	$= N/2^2$
3	→	$2/8$	$= N/2^3$
⋮		⋮	⋮
k steps	→	1	$= N/2^k$

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

$$2^k = N$$

$$\log_2 2^k = \log_2 N$$

$$k \log_2 2 = \log_2 N$$

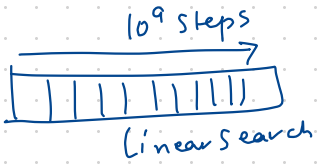
$$k = \log_2 N$$

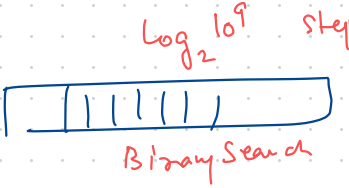
B.S. in worst case takes $\log N$ steps

$$\log_2 (2) = 1$$

(sorted)

Array Size $\rightarrow 10^9$





$= \log_2 (10^3)^3$

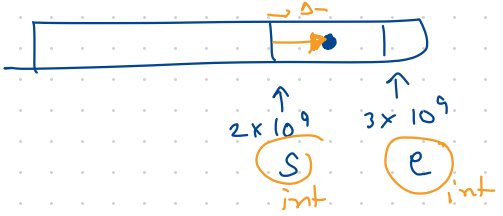
$=$

$\frac{3 \log_2 1000}{} = 3 \times 10^{-7}$ seconds.

$= 30$ steps

1s $\rightarrow 10^8$ ops

Time $\frac{10^9}{10^8} = 10$ seconds



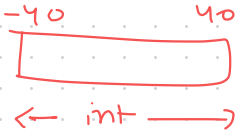
too big $\rightarrow$ beyond range of int sometime

$mid = \left(\frac{s+e}{2}\right)$

small no

$$mid = s + \frac{(e-s)}{2} \Rightarrow \frac{2s+e-s}{2} = \left(\frac{s+e}{2}\right)$$

more error proof



$s = 30 \qquad e = 40$

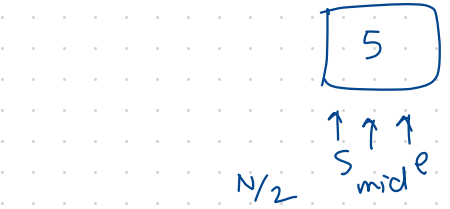
$mid = \frac{30+40}{2} = \frac{70}{2} = 35$ ← otherwise

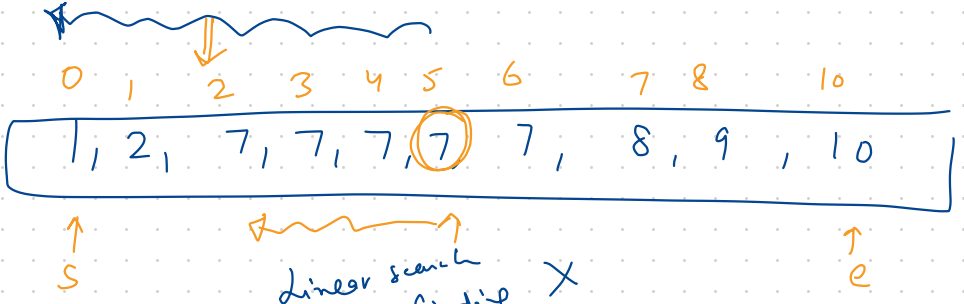
$mid = 30 + \frac{(40-30)}{2}$

$= 30 + \frac{10}{2}$

$= 30 + 5 \quad = 35$

10.40





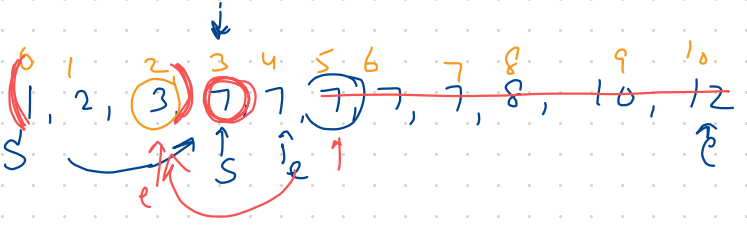
Key = (7)

mid = $\frac{0+10}{2}$ = (5)

any Random occ.

$\propto$ N iterations.

First occ



store & go left

0 – 10 mid = $\frac{0+10}{2}$ = (5)

ans = 5 3

0 – 4 mid = $\frac{0+4}{2}$ = (2)

3, 7

3 – 4 mid = $\frac{3+4}{2}$ = (3)

update ans

e = mid – 1

[e = 2
[s = 3 ⟶ Stop

Think: Given a Number N, find out the square root N without using pre-built method

20 → 4 (int part)

25 → 5

49 → 7

110 → 10

- Linear Search
- Binary Search.

Brute Force / Basic

li ~

N = 30

↓

Sq is

<= 30

N

↓

i = 0 0^2 <= 30

i = 1 1^2 <= 30

i = 2 2^2 <= 30

i = 3 4^2 <= 30

i = 4 5^2 <= 30

i = 5 6^2 <= 30 ⇒ stop

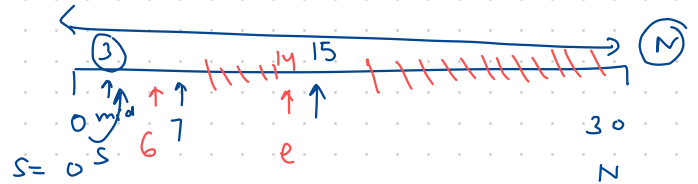
Root N

i = 0 ——— i <= $\sqrt{N}$

i - 1

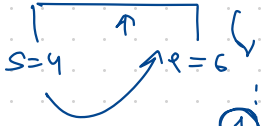
i > $\sqrt{N}$ Stop

Binary Search



ans = 3 5

N	0-30	mid = 15	$15^2 <= 30$	NO	Left
N/2	0-14	mid = 7	$7^2 <= 30$	NO	Left
N/4	0-6	mid = 3	$3^2 <= 30$	Yes	Right
	4-6	mid = 5	$5^2 <= 30$	Yes	Right
1	6-6	mid = 6	$6^2 <= 30$	NO	
	s=6, e=6	Stop			



$\sqrt{10}$ = 3.162

N: 0 to 9 | 0-9 | 0 to 9

3 places

→ Linear Search for each digit (9)

→ Binary Search for each digit

$3.0^2 <= 10$

$3.1^2 <= 10$

$3.2^2 \not<= 10$

$2^3\ 2^2\ 2^1\ 2^0 \quad 2^{-1}\ 2^{-2}\ 2^{-3}\ 2^{-4}\ 2^{-5}$

$1\ 0\ 1\ 0 . \quad 1\ 0\ 1\ 0\ 1 \ldots$

11

0.5 0.25 0.125

Negative No's

5.25

101

3.162

BS

$\log N + p * 10$ (constants)

$\log N + p * \log 10$

3.16 20000

3.16 19999 ...

3.162

Monotonic Seard Spaces