

Quick Sort → Recursion

$$\text{arr} = \{ \underline{50}, 40, 80, 90, 70, \underline{10}, 30, 60, 120 \}$$

B.P = Sort n elements

S.P = Sort $\binom{n-1}{1}$ element | S.W

Place 1 element at its correct position

$$\{ \underline{50}, 40, 80, 90, \underline{70}, \underline{10}, 30, 60, 120 \}$$

↓

$$\{ \underline{50}, 40, \underline{80}, \underline{90}, \underline{70}, \underline{10}, \underline{30}, \underline{60}, \underline{120} \}$$

$$\{ 10, 40, 50, 80, 90 \}$$

$$\{ \underline{30}, \underline{60}, \underline{120} \}$$

① Place 70 at correct posn

② Elements < 70 should be on the left

③ Elements > 70 should be on the right

All elements smaller than pivot must be on the left
all elements larger than the pivot must be on the right

Steps :
 1) Find Pivot
 2) Partitioning $\xrightarrow{\text{left}} \xrightarrow{\text{right}}$
 3) Recursion

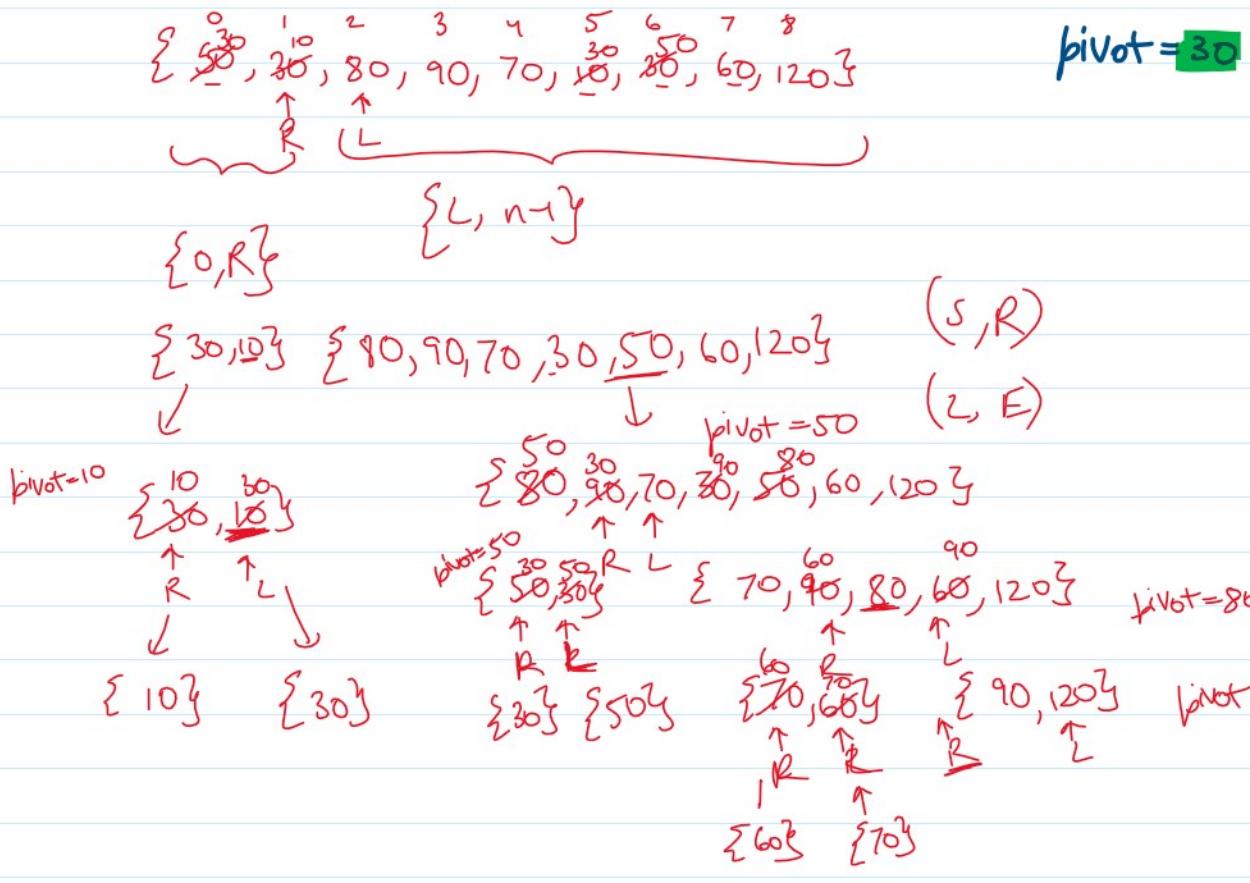
$$\{ \underline{50}, 40, 70, \underline{10}, \underline{30}, \underline{90}, \underline{10}, \underline{70}, \underline{30}, \underline{60}, \underline{120} \}$$

equality is also a problem

$$\{ \underline{30}, \underline{10}, 80, 90, 70, \underline{30}, \underline{80}, \underline{50}, \underline{60}, \underline{120} \}$$

Inplace $\rightarrow \times$

pivot = 30



quickSort (arr, s, c) {

$\text{pivot} = -X \rightarrow$ har funk'n frame ka different pivot hoga
 $L = S, R = e$

// partition

quickSort(Larr, S, R);
quickSort(Larr, L, E);

1

Base case

1) $s > e$

return;

{70y
L R



$$T(n) = T(n) + T(n-n) + \underbrace{n}_{\rightarrow \text{Partition}}$$

pNot = arr[mid]

How to choose pivot??
1) $n=1/n-1 \rightarrow$ pivot is smallest/largest

2) $n = n/2 \rightarrow$ Pivot is the Median of the arr \times

How to choose pivot??

1) $n=1/n-1 \rightarrow$ Pivot is smallest/largest

$$T(n) = T(1) + T(n-1) + n$$

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

$$T(n-2) = T(1) + T(n-3) + (n-2)$$

$$T(n-3) = T(1) + T(n-4) + (n-3)$$

$$\vdots \\ T(2) = T(1) + T(1) + 2$$

$$T(1) = 1$$

$$T(n) = n \cdot T(1) + n + (n-1) + (n-2) + \dots + 3 + 2 + 1$$

$$= n + n \frac{(n+1)}{2}$$

$$T(n) = O(N^2)$$

Worst Case

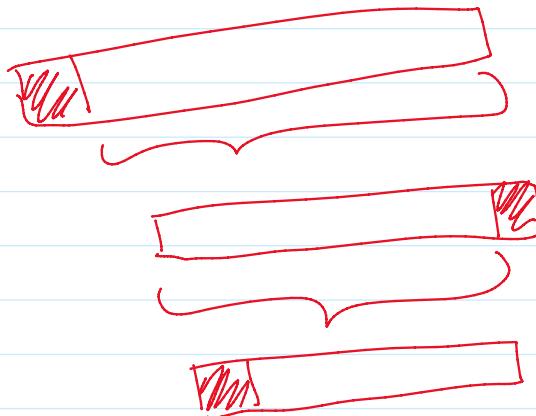
Pivot is the \times
Median of the arr

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

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

$O(N \log N)$

avg case / best case



$$l < r$$

$\{ 20, 30, 50, 30, 70 \}$ pivot = 35

$\{ 60 \}$
 $l \quad r$

S, R

L, C

Stable Sorting Algorithm

{ 10, 20, 10₂, 5 }

↶ { 5, 10₁, 10₂, 20 }

Org Order of elements is retained

→ Merge Sort

{ 10¹, 5, 10² }

{ 10, 5 }, { 10² }

{ 5, 10³ } ↓ { 10³ }

{ 5, 10³ } ✓

Bubble Sort ✓

Selection ✓

Insertion { 5, ..., 5 }

5, 10², 10 ✓

Quick Sort

{ 5⁰, 30, 10¹, 80, 30², 10², 70, 10³, 90, 30⁴, 80⁵, 120 }

L R

{ 50, 30, 10¹, 30², 60³ } ↑↑
R L

{ 30⁰, 10¹, 30², 30³ } ↑
R L

{ 10, 30², 30¹, 50 }
↷ Unstable

10⁵ N²

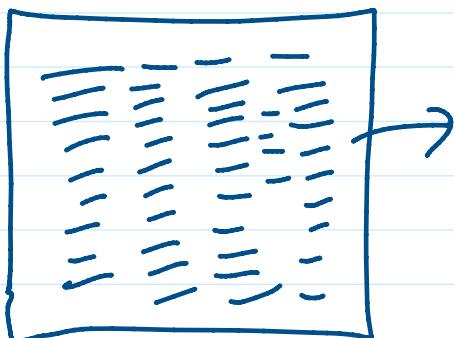
↷ n - - - - -

Bubble | Set | Ins | Merge | Quick

1) Almost sorted Array?

	Doubble	STL	MS	Range	Quicks
1) Almost sorted Array?			✓		
2) Kuch Nahi Pata?				✓	
3) Fix No of elements? 500	✓	✓			
4) Real Life?			✓		

Internally \Rightarrow Arrays.sort();
 $O(N)$
= Merge + Insertion ka mixture
↳ Stable Sorting Algo



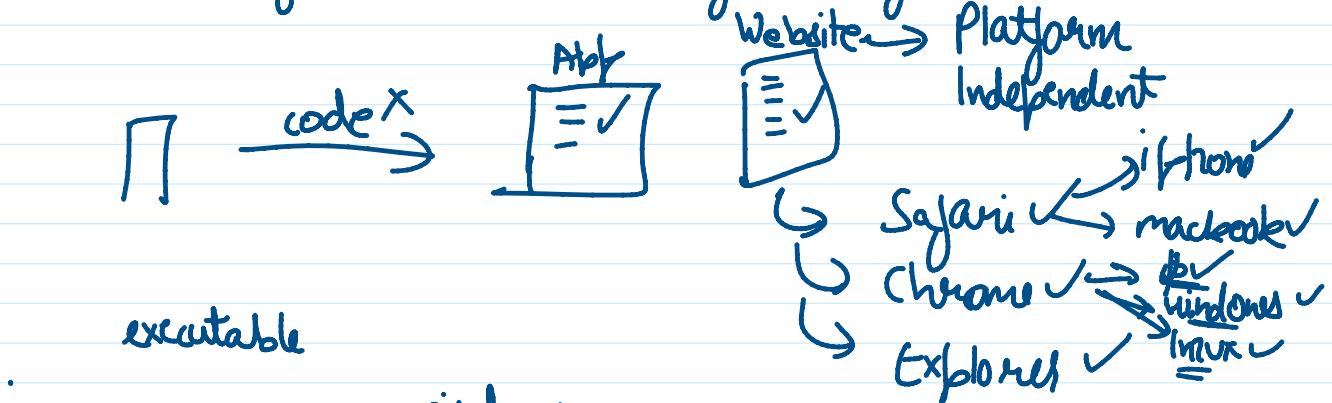
Kadane such that
subarray size > 4

Generalization

Chat GPT X

Reinforcement

OOPS → Object Oriented Programming



GTA → windows
exe

Samsung Galaxy Ace
HTC Explorer ✓ X

Symbian

C++ → Executable ✓
.exe

Jawab X

Java → Window
byte class → Mac
byte class → Linux
byte class → Phone
byte class → PS5



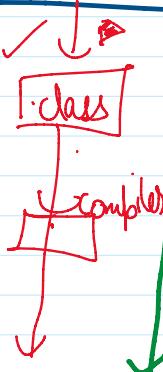
byte class

Code , High level language

english
Fridge .onc)

JVM runtime pc
.class ke machine
level code mein
convert krdeta hai

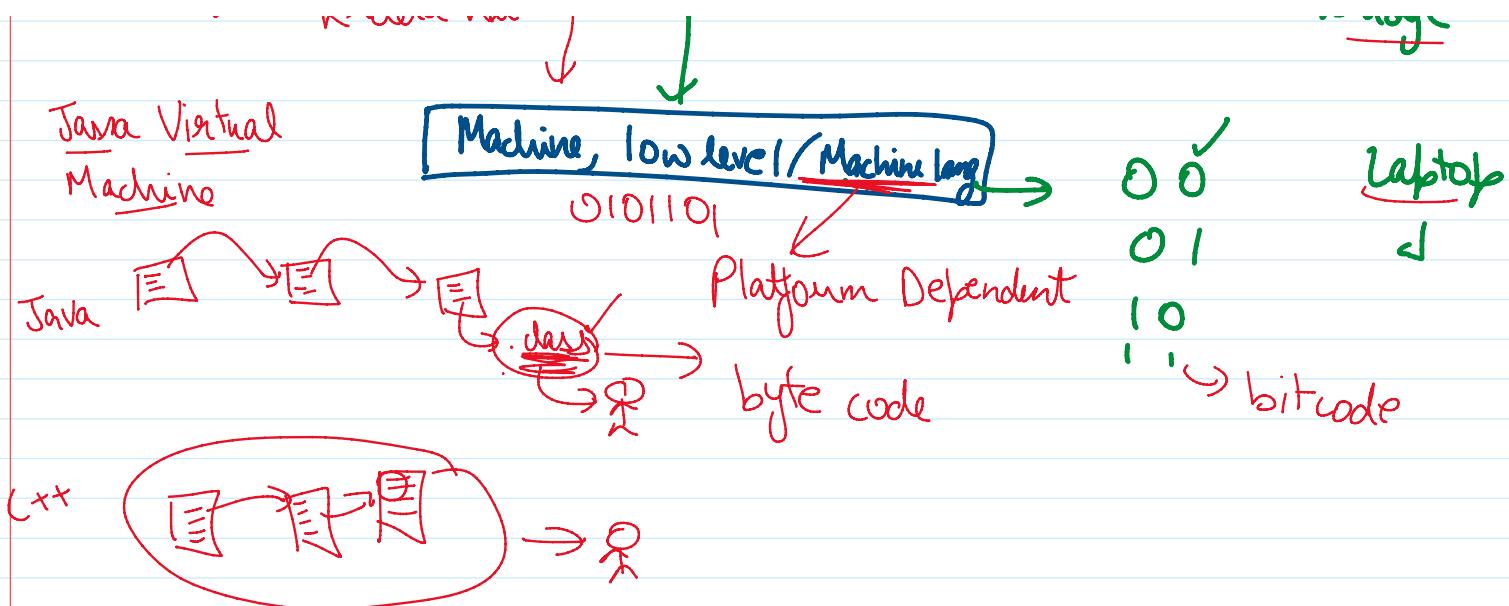
Compiler



Compiler

HLL convert to LLL

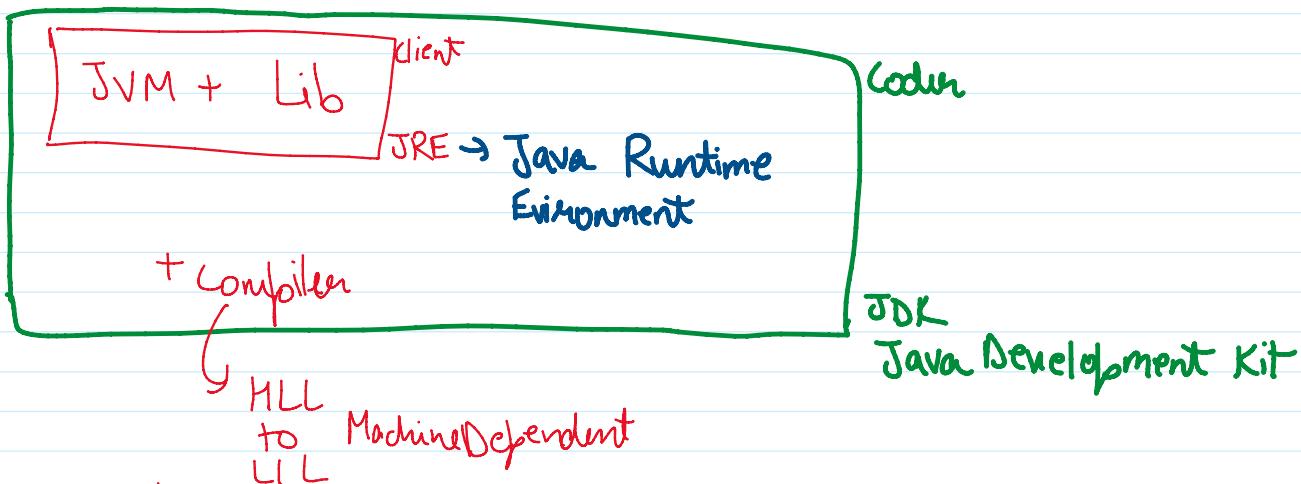
Fridge



	Client	Coder (Amazon)
Compiler	X	✓
JVM	✓	✓
Library	✓	✓

↳ Code mein jo likha usko import karte hai

ArrayList AL = new ArrayList();
↳ Library??



Client needs JRE
Coder needs JDK

Kisi bhi machine mein JRE hua to java ka code chal jayega