

- Kadane's Algo
- Array list
- Double Set

Maximum Sum Subarray

[3, 4, -2, 0]

sum = 0

[3]

[3, 4]

[3, 4, -2]

[3, 4, -2, 0]

ans = 0;

for (i=0; i<n; i++)

sum = 0;

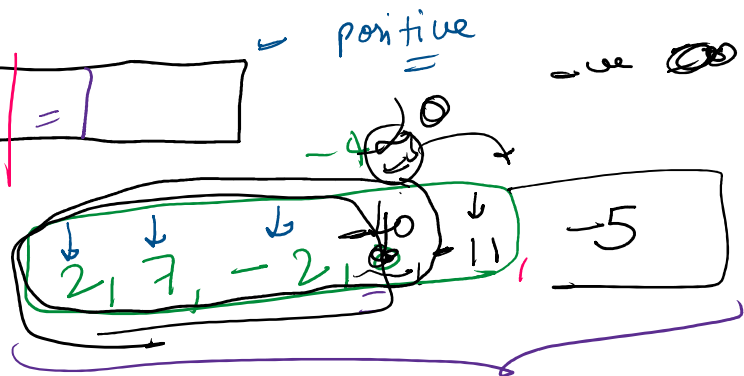
for (int j=i; j<n; j++)

sum += arr[j];

if (sum > ans) ans = sum;

9 = 12

sum = 2



Array list

→ problem with arrays is - size is fixed.

[CROD]

→ Arrays

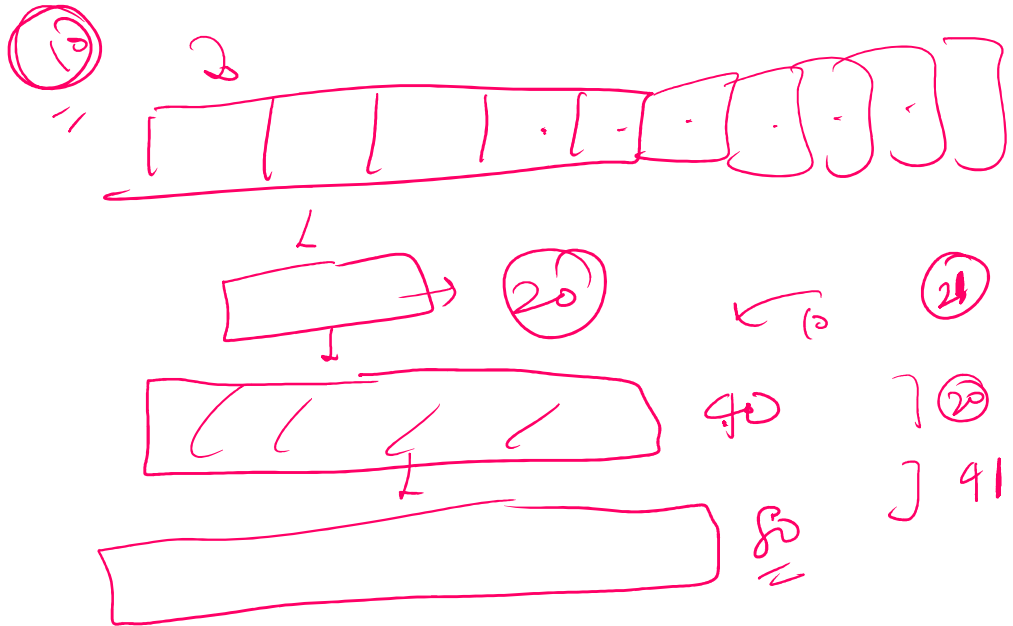
int a[] = new int [10];
a[5] = 15;



→ ArrayList is a dynamic sized array

Memory Management of ArrayList

ArrayList<Integer> arr
= new ArrayList<>();



[1-1000]

[1-10] → O(1)

(11) → [1-10] → O(N)

O(1) → 1ms

10
20
10

119
1000
1.683

923X1 + 7X100

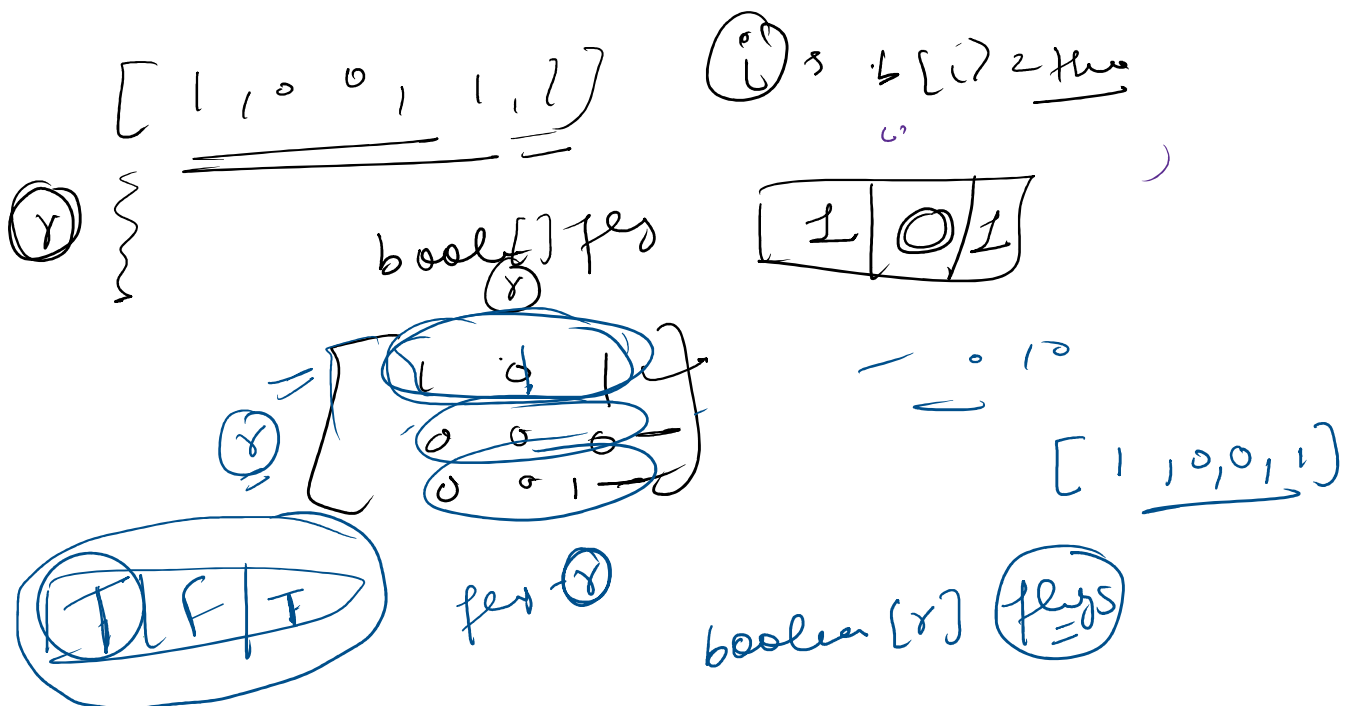
$O(1) \rightarrow 1 \text{ ms}$
 $O(N) \rightarrow \text{terms}$

$$\begin{array}{r} 20 \\ 40 \\ 80 \\ 160 \\ 320 \\ 640 \end{array} \left| \begin{array}{l} 225X + 1X^2 \\ \hline 10 \end{array} \right.$$

Time Complexity

$O(1) \rightarrow \text{insertion}$
 $O(1) \rightarrow \text{Read}$
 $O(1) \rightarrow \text{Update}$
 $O(1) \rightarrow$

get(2)



Sorting

2, 1, 4, 3

↳ 1, 2, 3, 4

Bubble Sort

(1, 3, 7, 2, 9)

[1, 3, 2, 7, 9]

(3, 1, 9, 7, 2)

(1, 3, 9, 7, 2)

1, 3, 7, 9, 2

(1, 3, 7, 2, 9)

(20, 20, 40, 10) = 1
20, (30, 10, 40)
20, 20, 10, 40

(20, 30, 10, 40) = 2
20, 10, (30, 40)
(20, 10, 30, 40)
(10, 20, 30, 40)

Complexity Analysis of Bubble sort

Worst case $O(N^2)$ } \Rightarrow reverse sorted

Space - $O(1)$

Best case - $O(N)$ \rightarrow already sorted

Avg case - $O(N^2)$

40, 30, 20, 10
↓
30, 20, 10, 40
↓
20, 10, 30, 40
↓
10, 20, 30, 40

hey ... 10, 20, 30, 40

↓
20, 10, 30, 40