

Merge 2 sorted Array.

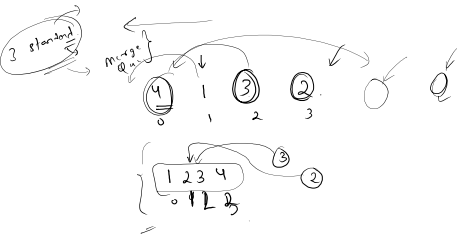
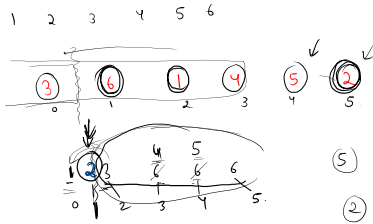
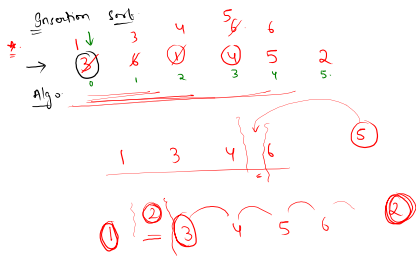
→ 1 3 5 7 ✓
 2 4 6 ✓

⇓

(7)

1 2 3 4 5 6 7

o(n)

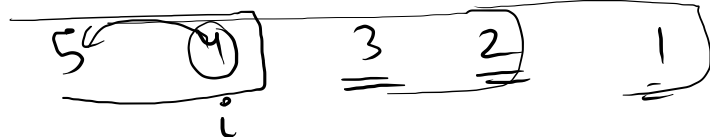


Aim: every it. largest end.



5 1 4
1 5 7

eg.



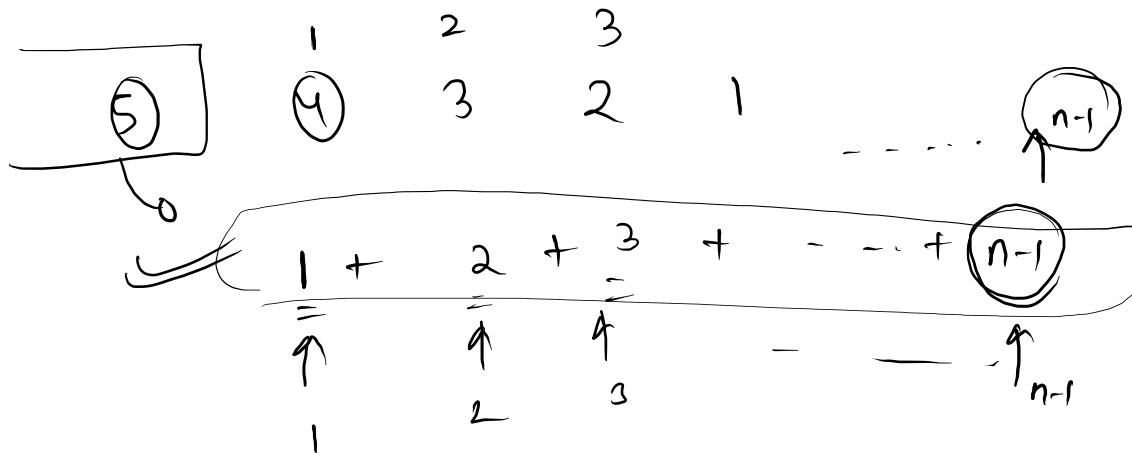
$$1 + 2 + 3 + 4 + \dots + n-1$$

$$1 + 2 + 3 + 4 + \dots + n-1 + \underline{\underline{n}}$$

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

$$\frac{(n-1)(n)}{2}$$

$$\approx \underline{O(n^2)}$$



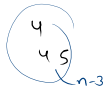
$$\underline{1 + 2 + 3 + 4 + \dots + (n-1)}$$

$$\begin{aligned}
 1 + \dots + n &\Rightarrow \frac{n(n+1)}{2} \Rightarrow \frac{(n-1)(n)}{2} \\
 &= \frac{n^2 - n}{2} \\
 &= O(n^2).
 \end{aligned}$$

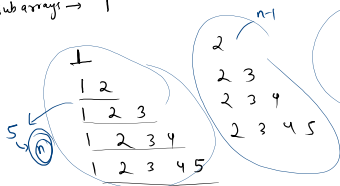
Print All subarray?

$n=5$

1	2	3	4	5
---	---	---	---	---

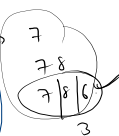
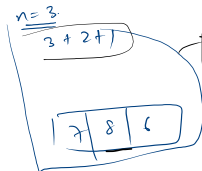


subarray \rightarrow 1



$n=5$

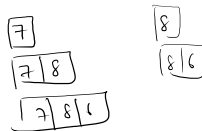
$$5 + 4 + 3 + 2 + 1$$



$n=4 \rightarrow 4 + 3 + 2 + 1$

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

Total $\Rightarrow \frac{n(n+1)}{2}$



$st=0$

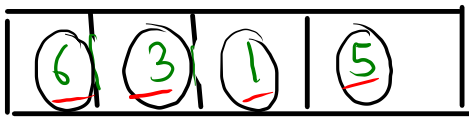


$st=1$



$st=2$





$\overline{\uparrow} = \uparrow$

st $\rightarrow [0, n-1]$

obs

st = 0
 end \rightarrow 0
 1
 2
 3

st = 1 \rightarrow 1
 2
 3

st = 2 \rightarrow 2
3

st = 3 \rightarrow 3

Kunal \rightarrow dry run.

