- Exam 1 scores

- Starting March 13, our lectures will begin at 7:30 am IST.

Divide and Conquer

Integer Multiplication.

Input: two integers X, Y.

X & Y both have n digits.

Obj: To multiply X & Y, i.e; to compute $X \cdot Y$.

Example:

$$\left.\begin{array}{l} X = 3\ 5\ 8\ 6 \\ Y = 4\ 7\ 9\ 1 \end{array}\right\} n = 4$$

$$\begin{array}{r} 3586 \\ \times\ 4791 \\ \hline \end{array}$$

$$
\left.\begin{array}{r}
3\ 5\ 8\ 6 \leftarrow n \\
+\ 3\ 2\ 2\ 7\ 4\ 0 \\
+\ \underline{\qquad\qquad} \\
+\ \underline{\qquad\qquad}
\end{array}\right\} n \qquad \Theta(n^2).
$$

Can we do better?

X_0, Y_0 : lower order $n/2$ digits of X & Y, respectively.

X_1, Y_1 : higher order $n/2$ digits of X & Y, respectively.

$$X = X_1 \cdot 10^{n/2} + X_0$$

$$Y = Y_1 \cdot 10^{n/2} + Y_0$$

In our example,

$X_0 = 86$, $X_1 = 35$, Clearly,

$3586 = 35 \times 10^2 + 86.$

$$XY = (X_1 \cdot 10^{n/2} + X_0)(Y_1 \cdot 10^{n/2} + Y_0)$$

$$\Rightarrow = \underbrace{X_1 Y_1}_{T(n/2)} \cdot 10^{n} + (\underbrace{X_1 Y_0}_{T(n/2)} + \underbrace{X_0 Y_1}_{T(n/2)}) 10^{n/2} + \underbrace{X_0 Y_0}_{T(n/2)}$$

Analysis :

$T(n)$: worst case running time of multiplying two n-digit integers.

Runtime recurrence.

$$T(n) = \begin{cases} O(1), & n = 1 \\ \underset{2}{4}T(n/2) + cn \end{cases}$$

3

Simplified Master Theorem.

Let $a \geq 1$ and $b > 1$, $k \geq 0$, be constants and let $T(n)$ be a recurrence of the form

$$T(n) = a\,T\left(\frac{n}{b}\right) + \Theta(n^k)$$

defined for $n \geq 0$. The base case $T(1)$ can be any constant value. Then

Case I : if $a > b^k$ then $T(n) = \Theta\left(n^{\log_b a}\right)$

Case II : if $a = b^k$ then $T(n) = \Theta\left(n^k \log_b n\right)$

Case III : if $a < b^k$ then $T(n) = \Theta(n^k)$.

For our recurrence

$a = 4, \quad b = 2, \quad k = 1$

Case I $\Rightarrow$ $T(n) = \Theta(\underline{n^{\log_2 4}}) = \underline{\underline{\Theta(n^2)}}$.

Consider the multiplication of

$(X_1 + X_0)(Y_1 + Y_0) = \underline{X_1 Y_1} + \underline{(X_1 Y_0 + X_0 Y_1)} + \underline{X_0 Y_0}$.

$(X_1 Y_0 + X_0 Y_1) = \boxed{(X_1 + X_0)(Y_1 + Y_0)} - \underline{X_1 Y_1} - X_0 Y_0$

mult. of two $n/2$ digit nos.

P

Algorithm IM (X, Y)

1. if $n = 1$ then

   Done.

2. else

$T(n/2)$

$x_1 y_1 \leftarrow IM(X_1, Y_1)$

$x_0 y_0 \leftarrow IM(X_0, Y_0)$ $T(n/2)$

$P \leftarrow IM(X_1 + X_0, Y_1 + Y_0)$ $T(n/2)$

3. return $x_1 y_1 10^n + x_0 y_0 + (P - x_1 y_1 - x_0 y_0) 10^{n/2}$.

Runtime recurrence:

$$T(n) = \begin{cases} O(1), & n=1 \\ 3T(n/2) + cn, & \text{o.w.} \end{cases}$$

By Case I of the Simplified Master Theorem, we have

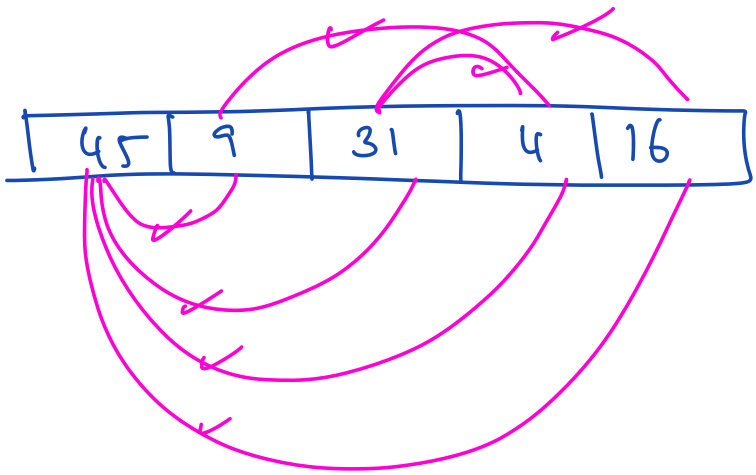
$$T(n) = \Theta\left(n^{\log_2 3}\right) = \Theta\left(n^{1.57}\right).$$

Counting Inversions.

Input: Array A of n distinct integers.

Obj: To count # inversions in A.

inversion happens when $i < j$, but
$A[i] > A[j]$



7 .

Algorithm

Sort and Count (A [1..n])

if $n = 1$ then

return (A , 0) $\Big\}$ $O(1)$

else

mid $\leftarrow \left\lfloor \frac{1+n}{2} \right\rfloor$ $\Big\}$ $O(1)$

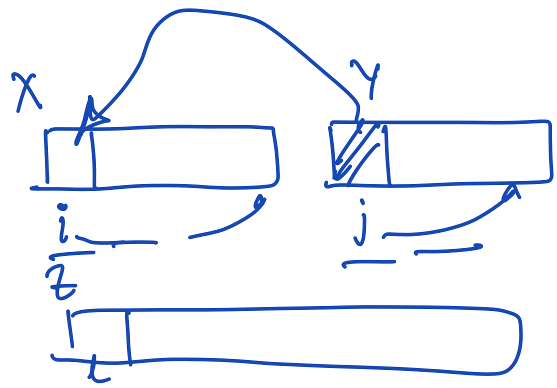
$T(n/2)$ * $(A_1, i_1) \leftarrow$ Sort and Count (A [1..mid])

$T(n/2)$ ⊛ $(A_2, i_2) \leftarrow$ Sort and Count $(A[mid+1, n])$

$O(n)$ ⊛ $(A', i_{12}) \leftarrow$ <u>Merge and Count</u> (A_1, A_2)

return $(A', \underline{i_1 + i_2 + i_{12}})$

Merge and Count $(\overset{n/2}{X}, \overset{n/2}{Y})$



$i \leftarrow 1$, $j \leftarrow 1$, $l \leftarrow 1$

$\#inv \leftarrow 0$

<u>while</u> $i \leq |X|$ and $j \leq |Y|$ <u>do</u>

<u>if</u> $X[i] < Y[j]$ <u>then</u>

$Z[l] \leftarrow X[i]$

l++

i++

<u>else</u>

$Z[l] \leftarrow Y[j]$

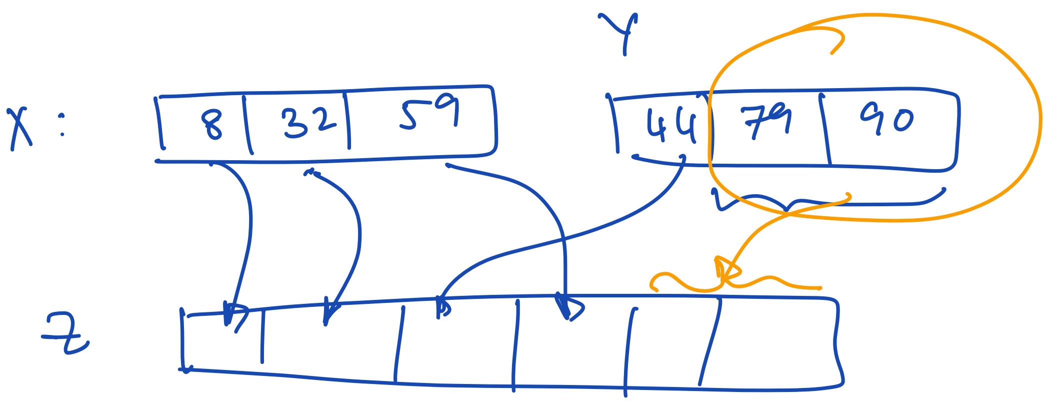
$\#inv \leftarrow \#inv + \underline{|X| - i + 1}$

ℓ ++

i ++

if $i < |X|$ then
    append $X[i, |X|]$ to Z.
else
    append $Y[j, |Y|]$ to Z.

return $(Z, \#inv)$.



Runtime recurrence.

$$T(n) = \begin{cases} O(1), & n = 1 \\ 2T(n/2) + cn, & \text{o.w.} \end{cases}$$

$$T(n) = \Theta(n \lg n)$$

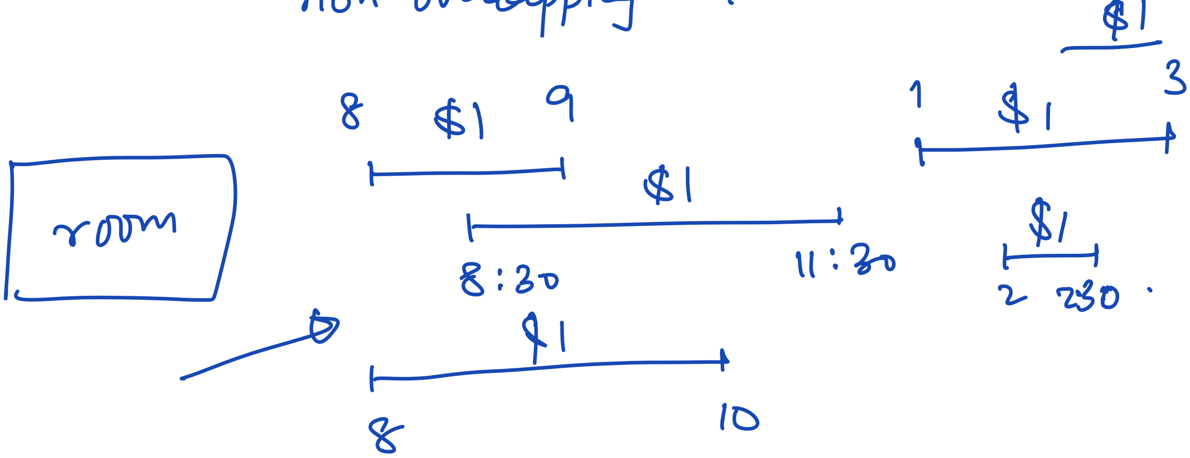
Greedy Algorithms.

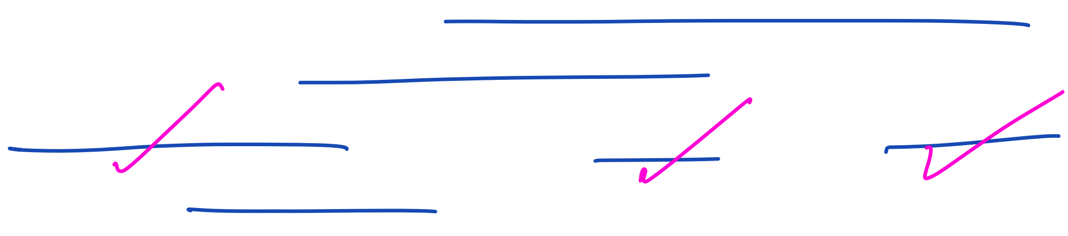
Interval Scheduling.

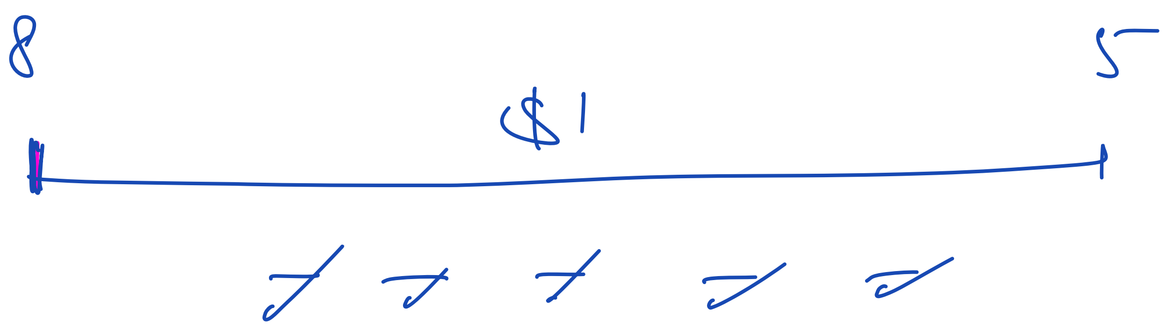
Input: n intervals: $1, 2, \ldots, n$

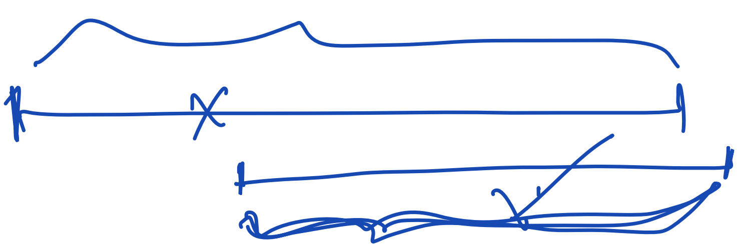
- interval i
  - start time s_i ✓
  - finish time f_i ✓

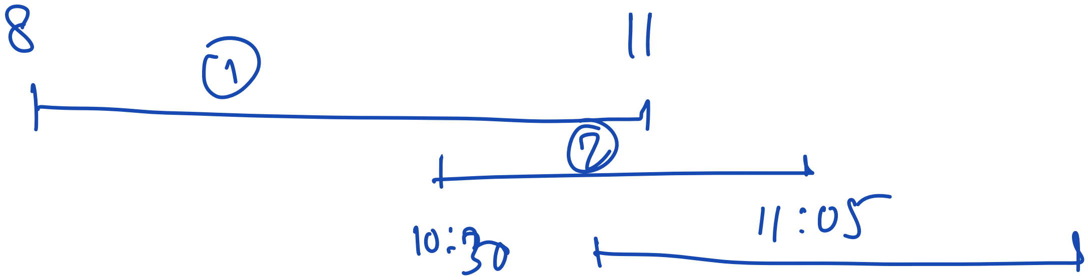
<u>Objective</u> : To find the max # non-overlapping intervals.










8
11
1
2
10:30
11:05