4/10/22

Divide and Conquer :

$n \to n^2$

"sorting"

selection sort n^2

P — n^2 : computational problem of size n.

P_1, P_2 - two sub-problems of size $< n$. ($= n/2$ each.)

$\left(\frac{n}{2}\right)^2 + \left(\frac{n}{2}\right)^2$ - ~~solve them~~

$= \frac{n^2}{2}$

$n^2/4$ - "combine" these two solutions to get the solution for P.

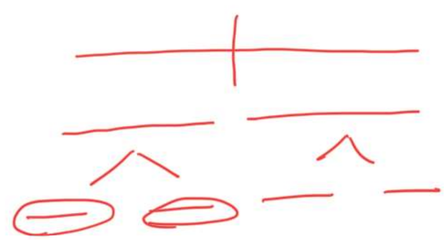
$< \frac{n^2}{2}$ time ? $< \frac{n^2}{2} + \frac{n^2}{2} = n^2$

Merge Sort / Quick Sort

Sort n elements {

Mergesort	Quicksort		
(i) Easy	(i) Non-trivial	(i)	Divide the input into 2 smaller inputs.
		(ii)	Recursively solve the two smaller inputs.
(iii) Non-trivial	(ii) Easy	(iii)	"Combine" the two solutions.

}.



1, 5, 6, 9 : 2 3 10 12

5 1 6 9 : 10 2 12 3

1, 5   6, 9   10 2 12 3

5 1   6 9

5 1   1, 5   6, 9   6, 9

| 5 | 1 | | 6 | 9 |

```
MergeSort( A, n) { if n=1 ✓
n/4   ✓A_1 ← First Half of A
3n/4  ✓A_2 ← Second Half of A
      MergeSort( A_1, n/2)
      MergeSort( A_2, n/2)
      A ← Merge (A_1, A_2)   ✓??
}.
```

n → $n/2$, $n/2$

A_1 A_2 $O(\ell_1+\ell_2)$.

A_1 ℓ_1

A_2 ℓ_2

x ?

Recurrence: $T(n)$: worst-case time to sort n elements ←

constant.

$$\boxed{T(n) \leq T(n/2) + T(n/2) + c\cdot n}$$

$T(n) \leq 2T(n/2) + O(n)$. ✓ Solution is $\boxed{O(n\log n)}$.

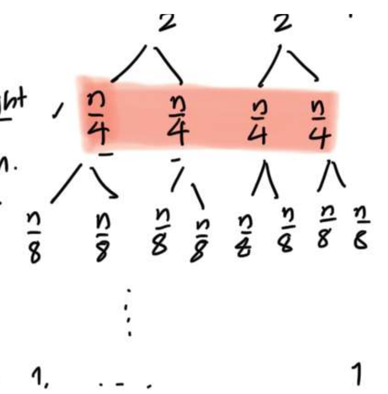
n

✓ n n + $c\cdot n$ (Merge cost) n $3n$

Time

$T(n) \le cn \cdot \underline{\text{Height}}$

$\le \underline{cn \cdot \log_2 n}$



$= \quad + c \cdot \left(\frac{n}{2}\right) + c\left(\frac{n}{2}\right) = \underline{cn}.$

$c\left(\frac{n}{4}\right) \cdot 4 = \underline{cn}.$

Examples:

$T(n/10) + T(9n/10)$

① $T(n) \le T(n/4) + T(3n/4) + cn.$

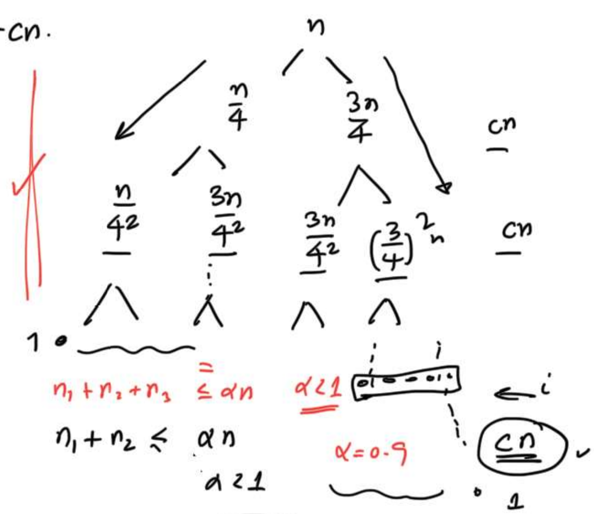
$= O(n \log_{4/3} n)$

$\left(\frac{3}{4}\right)^i n = 1$

Height $\le \log_{4/3} n$

$\underline{\log_{4/3} n} > \log_2 n$

$\log_2 n \left(\log_{4/3} 2\right) > 1$



$n_1 + n_2 + n_3 \le \alpha n \quad \alpha < 1$

$n_1 + n_2 \le \alpha n$

$\alpha < 1$

n-H

$\alpha = 0.9$

$T(n_1) + T(n_2) + T(n_3)$

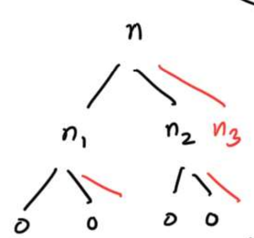
② $T(n) \le T(n_1) + T(n_2) + cn$

$T(n) = \; ? \; O(n)$

$cn + \alpha(cn) + \alpha^2(cn) + \dots$

$\frac{cn}{1-\alpha} = O(n).$

$\boxed{T(n) \le T(n/2) + cn}$



$+ \; cn_1 + cn_2 + cn_3 \le \alpha(cn)$

$\alpha^2(cn)$

$\alpha^3(cn)$

x

$T(n) = T(n/2) + O(1) \leftarrow T(n) = O(\log n)$

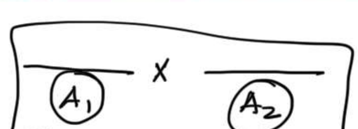
QuickSort(A, n) {

$x \leftarrow A[1]$

n_1 { $A_1 \leftarrow$ all elements less than x

n_2 { $A_2 \leftarrow$ all elements larger the x

Partition $c \cdot n$



Output

A_1, x, A_2

}

$n_1 + n_2 = n - 1.$

$T(n) \le T(n_1) + T(n_2) + cn$

$\sim n/4 \quad 3n/4$

$\boxed{1, 2, 3, \dots, n}$

$n + (n-1) + (n-2) + \dots$

Fix this? pick x "randomly"

Expected running time?

2, 3, ..., n — n

3, ..., n — $n-1$

H = n — n^2

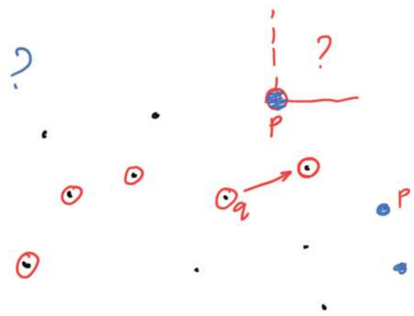
$\sim n^2$

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

(1) Maximal points. $O(n\log n)$?

Input: set of n points on the plane

A point p (x_p, y_p) is maximal if there is no other point p' with $x_{p'} \geq x_p$ and $y_{p'} \geq y_q$



Report all the maximal points — n^2 time.

for each point p check if p is maximal $O(n)$

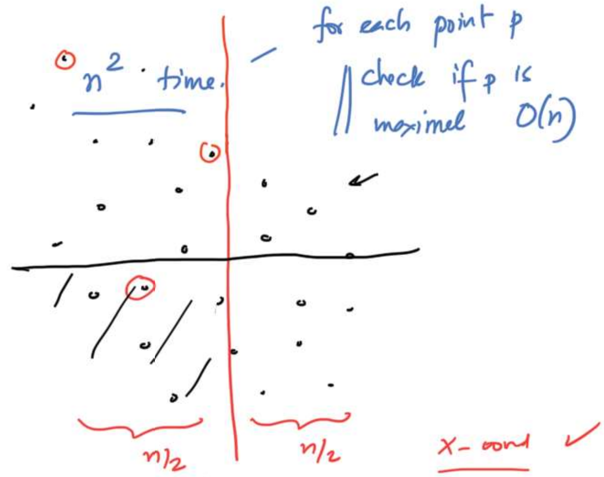
$O(n\log n)$ time?

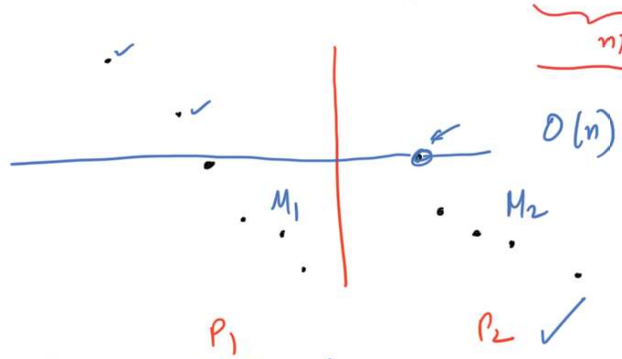
$T(n) = 2T(n/2) + O(n)$ ← $O(n)$

based on the "middle" x-coordinate

?? — P_1, P_2

Solve the problem P_1, P_2 recursively





$T(n) \leq 2T(n/2) + O(n)$.

Combined answer??

M_2: max. pts in P_2

M_1: max. " P_1

Add P_2 to M.

Find out the highest y-coord. in M_2 — q

M_1' ← set of points in M_1 with y-coord. higher than q

Add M_1' to M.

For — sort the points on x-coord. $n\log n$.

— for each recursive call, we will maintain the sorted points.

$n\log n$

Maximal (P, n) { ← points are in an array arranged by x-coord.

$p \leftarrow P[n/2]$ ←

$P_1 \leftarrow$ left half of P

$P_2 \leftarrow$ right half of P

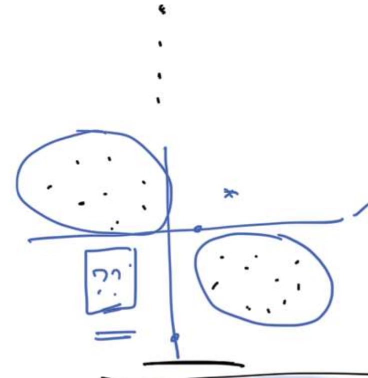
$M_1 \leftarrow$ Maximal $(P_1, n/2)$

$M_2 \leftarrow$ Maximal $(P_2, n/2)$

$M \leftarrow$ Combine (M_1, M_2)

return M

}.



② Closest Pair of points : $O(n\log n)$ time? $\boxed{T(n) = 2T(n/2) + O(n)}$?

n points in the plane.

Find the closest pair of points.

n^2 time alg.

(x_1, y_1) (x_2, y_2) $O(n^2)$.

$(x_1 - x_2)^2 + (y_1 - y_2)^2$

arranged according to x-coord.

Divide into two equal parts based on the x-coord.

Closest (P, n) {

$P[1] \dots P[n/2]$

$P_1 \leftarrow$ left half

$P_2 \leftarrow$ right half

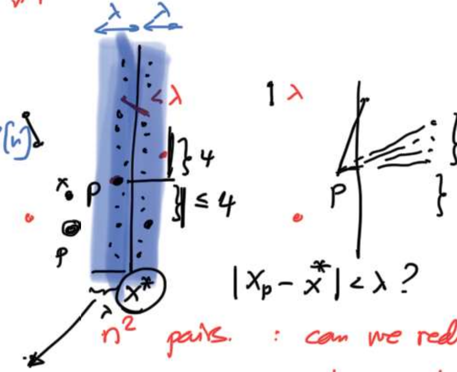
$P[n/2+1] \dots P[n]$

$p, p' \leftarrow$ Closest $(P_1, n/2)$

$q, q' \leftarrow$ Closest $(P_2, n/2)$

$\lambda_1 =$ dist (p, p'), $\lambda_2 =$ dist (q, q')

Output $\min(\lambda_1, \lambda_2)$??



$|x_p - x^*| < \lambda$?

n^2 pairs. : can we reduce this to n pairs only ??

only have to look at points which lie in this slab

$\Rightarrow O(n)$ pairs ?

$\lambda \leftarrow \min(\lambda_1, \lambda_2)$ ✓

P_1', P_2' : points of P_1, P_2 lying in the slab of width λ around x^*.

for each $p \in P_1'$ {

compare p with 4 points above it & 4 points below it in P_2'

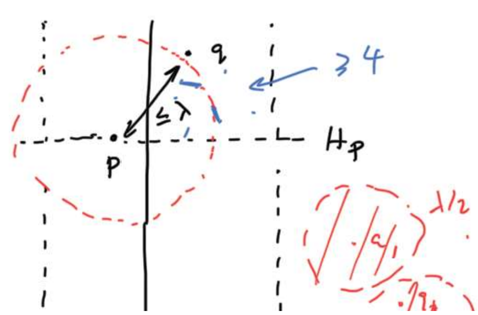
}

output the smallest dist. found.

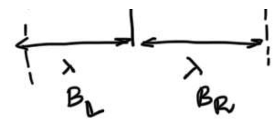
}

Fix a point $p \in B_L$.

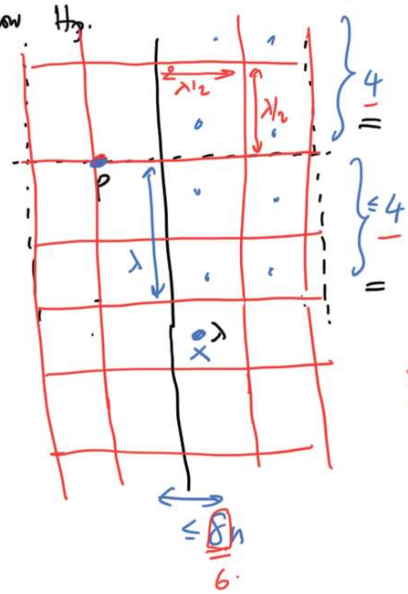
Consider all the points $q \in B_R$ dist $(p, q) \leq \lambda$. Then there are at most 8 choices for q, in fact,



≤ 4 points above H_y and ≤ 4 points below H_y.

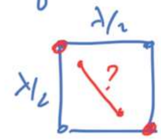


λ/λ



Consider squares of width $\lambda/2$

$O(n)$



$\frac{\lambda}{2} \cdot \sqrt{2} < \lambda$.

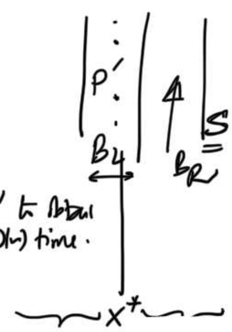
$? \; |y_p - y_q| < \lambda$ at most 8 such pts.

① P was sorted by the x-coord. We also need to maintain the point the points sorted by their y-coord.

P' Closest (P, P', n) { $O(n)$ time

sorted by x

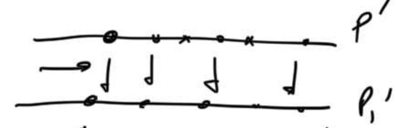
sorted by y-coord



$P_1 \leftarrow P[1] \ldots P[n/2]$, P_1' ← use P' to obtain P_1' in $O(n)$ time.

$P_2 \leftarrow P[n/2+1] \ldots P[n]$, P_2'

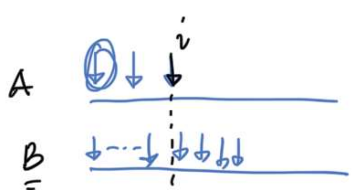
P': the points (x, y) sorted by y-coord.

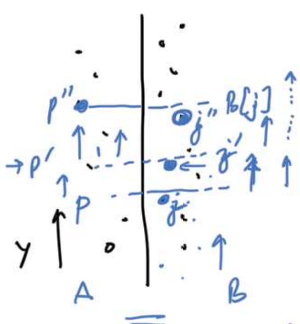


x^*: median x-coord

P_1': scan P' : (x, y) and $x < x^*$, add it to P_1'

Similarly we can obtain the points B_L, B_R sorted according to their y-coord.





B[j], B[j+1], ..., B[j-3]

B[j+1] ... B[j+4].

Merge.

$i = 0$, $j = 0$

repeat {

if $A[i] > B[j]$, copy $B[j]$ to C

j++,

A[i]

A:	1	5	10	15	17	
B:	2	3	7	9	12	13

else copy A[i] to C.
increment i

}.

$O(n \lg n)$, ✗

③ A : Find the median of A, $O(n)$ time.

$O(n)$ time.

A, k : Find the k^{th} smallest number (Selection).

$$T(n) = \boxed{T(n_1)} + \boxed{T(n_2)} + O(n) \quad \text{where } n_1 + n_2 \le c \cdot n$$

$T(n/5)$ where $c < 1$.

Select (A, k) {

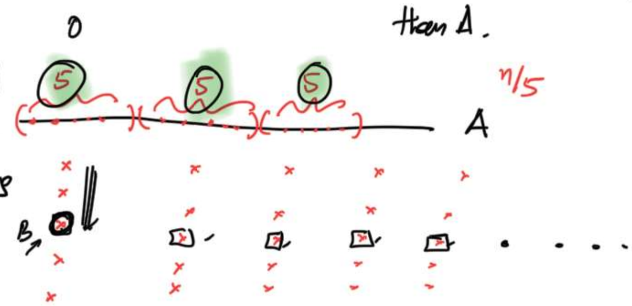
Select (A', k') ← A' will be a smaller array than A.

1. Break A into groups 5 consecutive elements. $O(n)$
2. Sort each such group. – $O(n)$ time
3. B ← new array formed by taking the middle element of

size of B? $n/5$.

ⓧ ← Median (B) Select (B, |B|/2) ✓

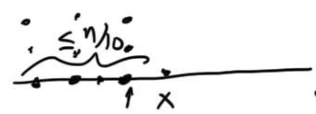
recursive call.



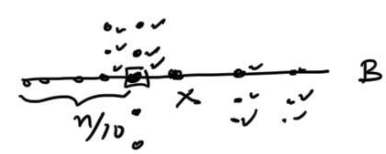
4. Partition the array A based on x

Claim: There are $\geq 3n/10$ — $n/3$ in A which are less than x. $T(n) = T(cn) + O(n)$ $O(n)$.

There are $\geq 3n/10$ in A which are more than x.



5. $\underline{\quad A_1 \quad}$ x $\underline{\quad A_2 \quad}$ $O(n)$ time.



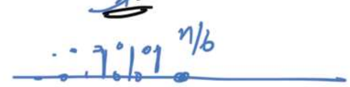
knowledge of x.

if $k < |A_1| - 1$ then
Select (A_1, k) ✓
if $k = |A_1| - 1$, output x
if $k > |A_1| - 1$
Select $(A_2, k - (|A_1| + 1))$

$$\boxed{T(n) = T(n/5) + T(7n/10) + O(n)}$$

$$\frac{7n}{10} + \frac{n}{5} = \frac{9n}{10}$$

$T(n) = O(n)$.



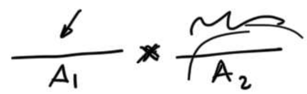
$\int \cdot$ $\cdot\, n/6 \cdot x$

Could we have formed gps of 3 instead of gps of 5?

$$T(n) = T(n/3) + T(2n/3) + O(n) \quad \Big\| \quad \frac{n}{3} + \frac{2n}{3} = n.$$

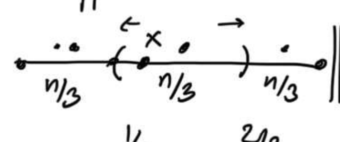
$O(n \lg n)$.

"Practical version"



T_n: running time of this alg. Expectation of T_n.

1. Pick a number x uniformly at random in A.
2. Follow ⊛ (Steps 4 and 5 using x).



3 times.

$1/3$. $2/3$

$\underline{Q}$: Suppose we have an experiment where the prob. of success is p. [How many times do we have to repeat it till we get a success? $\underline{\underline{X}}$

X: random variable.

$Pr[X=1]$, $Pr[X=i]$ $\qquad Pr[X=i] = (1-p)^{i-1} \cdot p$

$$\mathbb{E}[X] = \sum_{i=0}^{\infty} i \cdot Pr[X=i]$$

(Expectation of X)

Expected Value of X.

$$= \sum_{i=0}^{\infty} i \cdot (1-p)^{i-1} \cdot p$$

$$= 1/p$$

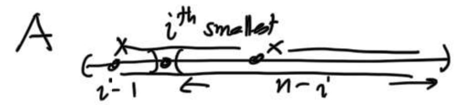
$Pr[H] = 1/2$ $\quad 1$

$Pr[T] = 1/2$ $\quad 0$

$\underline{\underline{1/2}}$.

$\mathbb{E}[T_n]$?

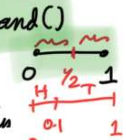
"sorted"



$\frac{1}{n}$. $\Big\|$ If we happen to pick the i^{th} smallest no. as x then the array recursive call would have size either $\underline{\underline{i-1}}$ or $\underline{\underline{n-i}}$

11/10/21. Selection Problem: A, k $\qquad k^{th}$ smallest

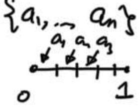
rand()



(i) Deterministic: fairly complex alg.

(ii) Randomized alg: uses randomness eg, $\|$ toss a coin

pick an element from $\{a_1, \ldots, a_n\}$ u.a.r.



Random Variable X: set of outcomes $\rightarrow \mathbb{R}$.

$\underline{\omega_1, \omega_2, \ldots, \omega_\ell} \rightarrow X(\omega_1), X(\omega_2), \ldots X(\omega_\ell)$

$p_1, p_2 \ldots p_\ell$

with each outcome there is

a prob.

$$\underbrace{\mathbb{E}X}_{\text{expectation}} = \sum_{\text{outcomes } \omega} p_\omega \cdot X(\omega).$$

Input I : execution of the algorithm is not fixed.

$\omega_1, \ldots, \omega_\ell$

executions of the alg. depending on the outcome rand().

$\underline{X}$: running time

$X(\omega_i)$: # steps in ω_i

$\mathbb{E}[X]$ is $\mathbb{E}$(running time) =

e.g., (i) Toss n unbiased coins. X : # heads. $\mathbb{E}X$.

$$\rightarrow \mathbb{E}X \hat{=} \sum_a a \cdot \Pr[X=a] = \sum_{a=0}^{n} a \cdot \binom{n}{a} \frac{1}{2^n} = n/2$$

$$\mathbb{E}[X] = \sum_{\boxed{\text{outcomes } \omega}} p_\omega \underbrace{X(\omega)}_{\overset{\shortparallel}{a}}$$

$$= \sum_a \sum_{\omega : X(\omega) = a} p_\omega \cdot a = \sum_a a \cdot \sum_{\omega : X(\omega)=a} p_\omega$$

$$= \sum_a a \cdot \Pr[X = a]$$

(1)

$$X = X_1 + X_2$$

X, X_1, X_2 are random variables.

Then,

$$\mathbb{E}X = \mathbb{E}X_1 + \mathbb{E}X_2. \checkmark$$

Linearity of Expectation.

It does NOT impose any conditions on X_1, X_2.

X, X_1, X_2 : take values in $\{0, 1, 2, \ldots, n\}$

$$\rightarrow \mathbb{E}(X_1 X_2) = \mathbb{E}(X_1) \mathbb{E}(X_2) \quad \times$$

Not true

$$\mathbb{E}X = \sum_a a \cdot \Pr[X=a]$$

$$= \sum_a a \cdot \Pr[X_1 + X_2 = a]$$

$X_1 = 1,\ X_2 = a-1$

$X_1 = 2,\ X_2 = a-2$

$\vdots$

$$= \sum_a a \cdot \sum_{b=0}^{n} \Pr[X_1 = b, X_2 = a-b] = \sum_a a \sum_{\substack{b, c \\ b+c=a}} \Pr[X_1 = b, X_2 = c]$$

$$= \sum^{n} \sum^{n} (b+c) \Pr[X_1 = b, X_2 = c] \checkmark$$

$b=0 \quad c=0 \quad = \quad$ ______ . ← incorrect $Pr[X_1=b] \cdot Pr[X_2=c]$ ✓

$$= \sum_{b,c} b \cdot Pr[X_1=b, X_2=c] + \sum_{b,c} c \cdot Pr[X_1=b, X_2=c].$$

$$= \underbrace{\sum_b b \cdot \underbrace{\sum_c Pr[X_1=b, X_2=c]}_{= Pr[X_1=b]}}_{\mathbb{E}X_1} + \cdots$$

$$\mathbb{E}X_1 + \mathbb{E}X_2.$$

$$X = (X_1+X_2)+X_3$$
$$\mathbb{E}X = \mathbb{E}(X_1+X_2) + \mathbb{E}X_3$$
$$= \mathbb{E}X_1 + \mathbb{E}X_2 + \mathbb{E}X_3$$

$$\mathbb{E}(X_1+\dots+X_n) = \mathbb{E}X_1+\dots+\mathbb{E}X_n.$$

Application: (i) n unbiased coins $\quad \mathbb{E}[\overset{X}{\# \text{heads}}] = n/2$?

Indicator Random Variables : 0 or 1 : $\mathbb{E}[Y] = 0 \cdot Pr[Y=0] + 1 \cdot Pr[Y=1]$
$= Pr[Y=1]$.

$$X_i : \begin{cases} 1 & \text{if the } i^{th} \text{ coin toss is H} \\ 0 & \text{if the } i^{th} \text{ coin toss is T} \end{cases} \qquad \mathbb{E}X_i = 1/2$$

$$\mathbb{E}[\#\text{ heads}] = \mathbb{E}[X_1] + \mathbb{E}[X_2] + \dots + \mathbb{E}[X_n]$$

Linearity of expectation

$$= n/2.$$

(ii) X random variable.

Suppose there is some other random variable Y. When $Y=i$; suppose $X=X_i$.

[throw a dice, if dice shows i, toss an unbiased coin i times.

$X = \#$ heads.

$$\mathbb{E}X = \sum_i Pr[\text{dice}=i] \; \mathbb{E}[\#\text{heads when we throw } i \text{ coins}].$$

$$= \frac{1}{6}\left[\frac{1}{2}+\frac{2}{2}+\dots+\frac{6}{2}\right]$$]

$$X = \begin{cases} X_1 & \text{when } Y=1 \\ X_2 & \quad " \quad Y=2 \\ \vdots & \\ X_n & \dots \quad Y=n \end{cases}$$

[Suppose Y is independent on the random variables]
$X_1, \dots, X_n$. $\Rightarrow Z_1, Z_2, \dots, Z_n$ are ind. of $X_1, \dots, X_n$

Then $\mathbb{E}X = \underbrace{Pr[Y=1]}\, \mathbb{E}X_1 + Pr[Y=2]\, \mathbb{E}X_2 + \dots + Pr[Y=n]\, \mathbb{E}X_n.$

Let $Z_i = \begin{cases} 1 & \text{if } Y=i \\ 0 & \text{otherwise.} \end{cases}$

$X = X_1 Z_1 + X_2 Z_2 + \ldots + X_n Z_n.$

$$\mathbb{E} X = \mathbb{E}(X_1 Z_1) + \mathbb{E}(X_2 Z_2) + \cdots + \mathbb{E}[X_n Z_n]$$
$$= \mathbb{E}[Z_1] \cdot \mathbb{E}(X_1) + \cdots + \mathbb{E}(X_n)\,\mathbb{E}(Z_n)$$
$$= \Pr[Y=1]\,\mathbb{E}(X_1) + \cdots + \Pr[Y=n]\,\mathbb{E}(X_n).$$

---

(11) Selection $(\overset{n}{A}, k)$ {

- pick an element x u.a.r. from A
- split A into

$$\underset{A_1}{\underline{<x}} \quad x \quad \underset{A_2}{\underline{\geq x}} \qquad O(n) \text{ time.}$$

- if $|A_1| = k-1$ ✓
- if $|A_1| \geq k$
  Selection (A_1, k)
- if $|A_1| < k-1$
  Selection $(A_2, k - |A_1| - 1)$

}

$T(n) = T(\alpha n) + O(n)$
$\alpha < 1$

X_n : Expected Running Time on an array of size n.

$\to T_n$: # steps on an array of length n.

$X_n = \mathbb{E}\, T_n.$

$\to T_n = T_{i-1}$ or $T_{n-i} + O(n)$ if the element x is the i^{th} smallest element

$$T_n \leq \underbrace{\max(T_{i-1}, T_{n-i})} + O(n) \quad \text{if } x \text{ is the } i^{th} \text{ smallest}$$

$A'[i]$

A' is the array A sorted in inc. order

$\to T_n \leq T_{\max(i-1,\, n-i)} + O(n)$ if x is the i^{th} smallest # in A.

$$\therefore \mathbb{E}\, T_n \leq \sum_{i=1}^{n} \left[\mathbb{E}\, T_{\max(i-1,\, n-i)} + O(n) \right] \cdot \frac{1}{n}$$

$$X_n \leq \underbrace{\frac{1}{n} \sum_{i=1}^{n} X_{\max(i-1,\, n-i)}} + O(n)$$

i

$$= \frac{1}{n} \left[X_{n-1} + X_{n-2} + X_{n-3} + \cdots + X_{n/2} + X_{n/2} + X_{n/2+1} + \cdots X_{n-1} \right] + O(n)$$

$$\to X_n \leq \frac{2}{n} \left[X_{n/2} + X_{n/2+1} + \cdots + X_{n-1} \right] + O(n) \leftarrow$$

$$\leq \frac{2}{n} \left[\frac{n}{4} \cdot X_{3n/4} + \frac{n}{4} \cdot X_n \right] \leftarrow$$

6 ~~2~~ 2 10

4 5 7

$+O(n)$ $\quad X_9 + X_8 + X_7 + X_6 + X_5 + X_5 + X_6 + X_7 + \dots + X_9.$

$X_n \le \frac{X_{3n/4}}{2} + \frac{X_n}{2} + O(n)$ ✓

$X(n) \le X(\alpha n) + O(n)$

$\alpha < 1$

$\boxed{X_n \le X_{3n/4} + O(n)}$ sol : $O(n)$

$X_i \ge X_{i-1}$. ✓

$\boxed{X_{n/2} + X_{n/2+1}\, X_{n/2+2}} \dots \boxed{\dots + X_n}$

$\underbrace{X_{n/2} + \dots + X_{3n/4}}_{\le \frac{n}{4} \cdot X_{3n/4}} + \underbrace{X_{3n/4+1} \dots + X_n}_{\le \frac{n}{4} \cdot X_n}$

(II) Randomized. Quicksort. $\quad n^2$

QuickSort(A, n) {

(i) choose $x \in A$ u.a.r.

(ii) Partition A:

$\underline{\quad A_1 \quad}\ x\ \underline{\quad A_2 \quad}$ $\quad O(n)$.

(iii) Quicksort$(A_1, |A_1|)$, Quicksort$(A_2, |A_2|)$.

}

$T_n = T_{i-1} + T_{n-i} + O(n)$ w.p. $\frac{1}{n}$ (ie, if x is the i'th smallest number).

$X_n = \frac{1}{n} \sum_{i=1}^{n} [X_{n-i} + X_{i-1}] + O(n)$

10

$X_n = \frac{2}{n} [X_1 + \dots + X_{n-1}] + O(n)$ ✓

$\begin{bmatrix} X_0 + X_9 \\ X_1 \quad X_8 \\ X_2 \quad X_7 \\ \vdots \\ X_9 \quad X_0 \end{bmatrix}$

$O(n \log n)$.

We will solve this and show that $X_n = O(n \log n)$. ✓

$T_n = [O(n) + T_{i-1} + T_{n-i}]$ $\frac{1}{n} \cdot$ prob.

x : ith smallest

$X_n = \mathbb{E}\, T_n$

$X_n = \frac{1}{n} \sum_{i=1}^{n} [O(n) + X_{i-1} + X_{n-i}]$

$= \underbrace{O(n)}_{cn} + \frac{2}{n} [X_1 + X_2 + \dots + X_{n-1}]$.

$X_1 + X_2 + \dots + X_{n-2}$.

$$X_n = cn + \frac{2}{n}\left[X_1 + X_2 + \dots + X_{n-1}\right]$$

$$X_{n-1} = c(n-1) + \frac{2}{n-1}\left[X_1 + X_2 + \dots + X_{n-2}\right].$$

$$X_n = cn + \frac{2}{n}\left[X_{n-1} + \frac{(X_{n-1} - c(n-1))(n-1)}{2}\right]$$

$$= cn + \frac{2X_{n-1}}{n} + \left(1 - \frac{1}{n}\right)X_{n-1} - \frac{c(n-1)^2}{n}$$

$$\frac{c(n^2 - 2n + 1)}{n} = c\left(n - 2 + \frac{1}{n}\right)$$

$$\le \left(1 + \frac{1}{n}\right)X_{n-1} + 2c$$

$$X_n \le \left(1 + \frac{1}{n}\right)X_{n-1} + 2c.$$ solve this!

$$= \left(1 + \frac{1}{n}\right)\left[\left(1 + \frac{1}{n-1}\right)X_{n-2} + 2c\right] + 2c.$$

$$= \left(1 + \frac{1}{n}\right)\left(1 + \frac{1}{n-1}\right)X_{n-2} + 2c\left(1 + \frac{1}{n}\right) + 2c.$$

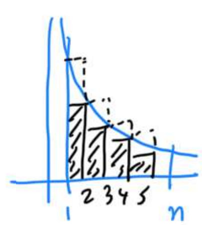
$$= \left(1 + \frac{1}{n}\right)\left(1 + \frac{1}{n-1}\right)\left(1 + \frac{1}{n-2}\right)X_{n-3} + 2c\left(1 + \frac{1}{n}\right)\left(1 + \frac{1}{n-1}\right) + 2c\left(1 + \frac{1}{n}\right) + 2c.$$

$i = n-1$

$$= \frac{n+1}{n-i}X_{n-i-1} + 2c\cdot\left[\frac{n+1}{n-1} + \frac{n+1}{n-2} + \dots + \frac{n+1}{n-i}\right].$$

$$= n + 2c(n+1)\underbrace{\left[\frac{1}{n-1} + \frac{1}{n-1} + \frac{1}{n-3} + \dots + 1\right]}_{\ln n} = n + 2cn\ln n.$$

$$\ln n - 1 \le 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} \le \ln n + 1$$



$y = \frac{1}{x}$

$\ln n - \ln 1 = \ln n.$

$$\boxed{\mathbb{E}[T_n] = O(n\log n).}$$

$\frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}.$

n^2

$1, 2, 3, \dots, n$

T_1

T_2

$\vdots$

T_ℓ

$$\boxed{\frac{T_1 + \dots + T_\ell}{\ell} \approx n\log n.}$$

$$\Pr[T_i \ge 100\, n\log n] \le 2^{-n}$$