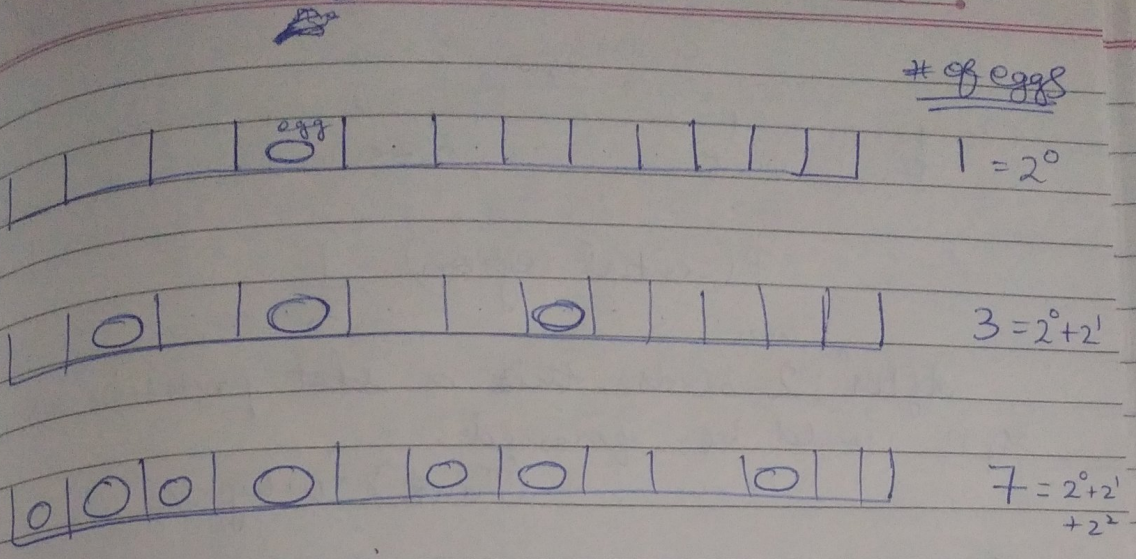


So on

each time the no. of eggs doubles (approximately)

bird 1 = 1 $\sim 2^1$

bird 2 = 2(1)+1 $\sim 2^2$

bird 3 = 2(3)+1 $\sim 2^3$

so on

the $\log_x$(no. of eggs) = birds

x^{birds} = no. of eggs.

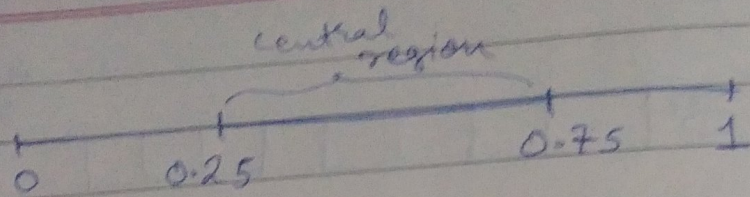
$x = ?$

(think!)

Rate at which slots are filling is exponential, so the number of birds should be ?

(like a binary tree)

(n leaves, means length of tree is ?)

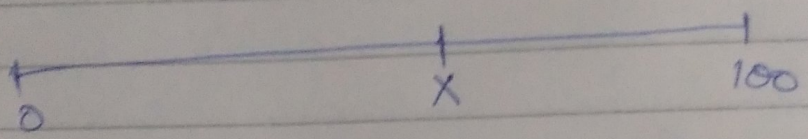


$$P(\text{central region}) = \frac{1}{2}$$

After at least 2 birds, ~~this~~ a ~~slot~~ partition of $\frac{3}{4}n$ would be formed.

(Think) → why $\frac{3}{4}^{th}$?

Math



$$F(X) = \begin{cases} \text{if } X \leq 50, & F(X) = 100 - X \\ \text{if } X \geq 50, & F(X) = X \end{cases}$$

$F(X)$	50	51	52	. . . . .	100
$P(F(X))$	$1/101$ → approx $= \frac{2}{101}$	$2/101$	$2/101$		$2/101$

$$E(F(X)) = \frac{2}{101}(50 + 51 - - - 100)$$

$$= \frac{2}{101}\left[\frac{100(101)}{2} - \frac{(50)(49)}{2}\right]$$

$$= 75.74 \approx \frac{3}{4}(100)$$

We only have to think when our biggest partition would fill up, because by then small partitions would already be filled. (think)

Math $\left(\frac{3}{4}\right)^{(\text{birds})} n \approx 1$

$$\left(\frac{3}{4}\right)^{\text{birds}} \approx \frac{1}{n}$$

$$\text{birds} \approx \log_{3/4}(1/n)$$

$$\text{birds} \approx \frac{-\log_{10} n}{\log(3/4)}$$

$$K \approx 8 \log n$$

$$K \approx \log n$$

'work' here means the number of comparisons we have to do for each iteration.

Date DELTA Pg No

Bird problem → k^{th} element

pivot is at $\left[\frac{3n}{4}\right]^{th}$ index of sorted list

we had to do 'n' work for 1st iteration
then $\frac{3}{4}n$ work for 2nd iteration,
then $\left(\frac{3}{4}\right)^2 n$ work until $\left(\frac{3}{4}\right)^k n$

here $k = \log n$.

Math

$$n, \frac{3}{4}n, \left(\frac{3}{4}\right)^2 n \; - - - \; \underbrace{\left(\frac{3}{4}\right)^k n}_{(\log n)^{th} \text{ iteration.}}$$

$$\text{Total work} = n\left(1 + \frac{3}{4} + \cdots \left(\frac{3}{4}\right)^{\log n}\right)$$

$$= n \frac{\left(1 - (3/4)^{\log n}\right)}{1/4}$$

$$= 4n\left(1 - (3/4)^{\log_{3/4}(1/n)}\right)$$

$$= 4n\left(1 - \frac{1}{n}\right) = 4(n-1)$$

$$= O(n)$$

Date
DELTA Pg No

Quick sort

Just like k^{th} element, but here in each step we will be doing n comparisons. i.e. work = n for each iteration.
(iterations)

~~There can be n! ways of inputting a list.~~
~~We can see this~~

no. of iterations = $\log n$

$\therefore$ Complexity = $n \log n$ (THINK!)

Why you cannot get better than $n \log n$?

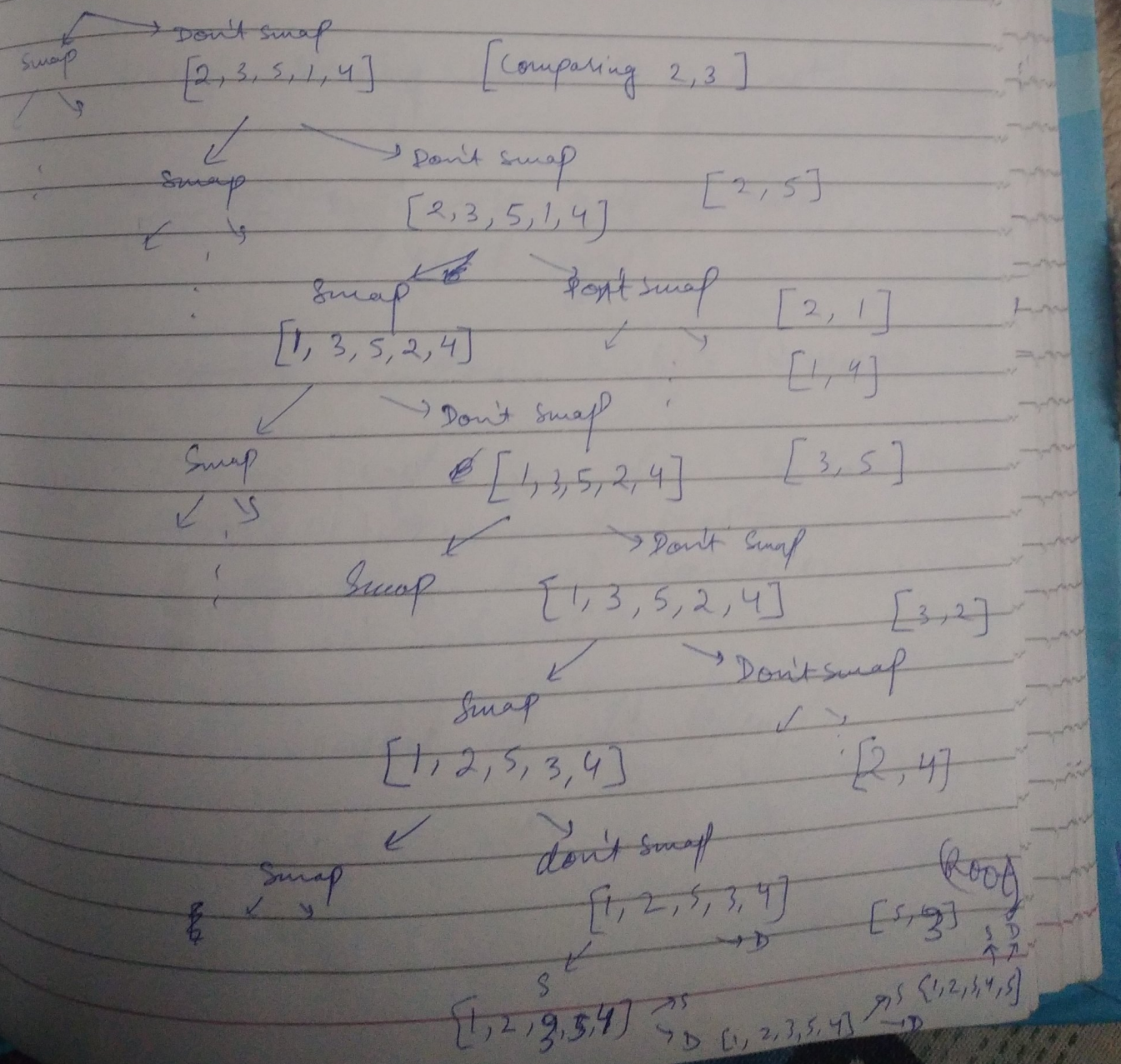
Think sorting as a binary tree, in which at each step you have 2 options (or possibilities for the list to change).

Bubble Sort as a tree

unsorted list → [Algo] → sorted list

- (len(list))! possibilities
- last level of tree has len(list)! leaves
- 1 possibility
- root of tree

[2,3,5,1,4] [out of 5! possibilities] (last leave)



This tree have 12 levels and 5! end leaves.

$\therefore$ height of tree $= \log_2 5! = 6.9 \approx 7 \neq 12$

But if n was really big, (THINK) → what would $\log n!$ be?

$$\log_2 n! = \log_2(n(n-1)(n-2)\cdots 1)$$

$$= \log_2 n + \log_2(n-1) \cdots \log_2 1$$

$$= \underbrace{\log_2(n) + \cdots \log_2(n/2)}_{\Downarrow} + \underbrace{\cdots \log 1}_{\text{negligible}}$$

$\because \log_2 n =$ no. of digit in 'n' in binary

$\therefore \log n! = \frac{n}{2} \log n$ (Think!)

Math

$$\frac{d(x \log x)}{dx} = 1 + \log x$$

$$\int 1 + \log x \, dx = x \log x$$

$$\int \log x \, dx = x \log x - x$$

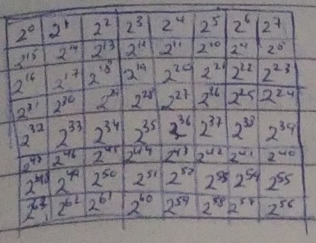
$$\log n! = \sum_{i=1}^{n} \log i \approx \int_1^n \log i \, di$$

$$= i \log i - i \Big|_1^n$$

$$= n \log n - n - \log 1 + 1$$

$$= n \log n - n + 1$$

$$= \theta(n \log n)$$



Chess board (intuition for exponentiation)

A monk came to a Kingdom. He challenged the King to play a game of chess with him. If monk wins, he would get rice grains, such that 1st square of chess board gets 1 rice grain, on 2nd square there are double the no. of rice grains on previous square and so on.

Last square would have 2^{63} rice grains i.e.

$9.22337204 \times 10^{18}$

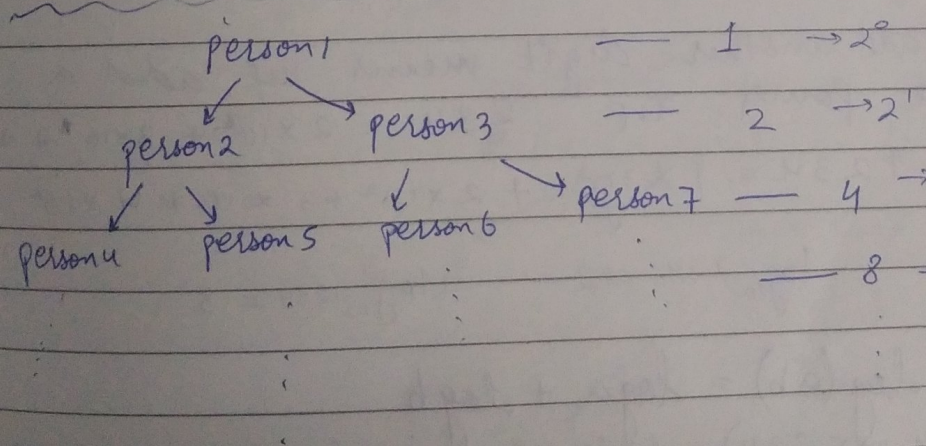
if 1g rice have 50,000 rice grains then 2^{63} rice grains means 1.8446×10^{14} kg

366 times

2021's rice consumption (of world) = 5043×10^{8} kg

1.844×10^{14} kg is just on the last square!

'Weird covid' spread



'weird covid' spreads, such that 1 infected person can only infect 2 non-infected people.

If I know 1000 people have weird covid (i.e. +ve tests) then how many levels does this spreading tree would have?

As we saw in the chess board example, the rice on last square are so much that ~~we can neglect rest~~ rice on other square can be neglected. So, we can approximate the number of people in tree as no. of people in the last level of tree.

The no. of people and levels are logarithmically (or exponentially) related.

$1000 \approx 2^k$

$k \approx \log_2 1000$

$\log_a b$ is approximately equal to no. of digits in 'b' when written in base 'a'.

$1000 \approx 1024 = 2^{10}$

$k = 10$

~~1000~~ 1000, in base 2 will actually have 9 digits, but we are approximating here.

we add another digit means we add a new power. i.e. $234 = 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$

$1234 = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10 + 4 \times 10^0$

$\log_{10} 234 \approx 2$ $\log_{10} 1234 \approx 3$

$\log(ab) = \log a + \log b$

$\log_{10}(1234 \times 234) = \log_{10}(1234) + \log_{10}(234) \approx 3 + 2 = 5$

$\log((2 \times 10^5) + \ldots) \approx 5$

#of digits in (ab) = (#of digits in a) + (#of digits in b)