Basics of Time Complexity

⇒ Agenda
- → String
- → Python Memory
- → Time Complexity

⇒ String →

String ≠ list of characters

"this is a string"

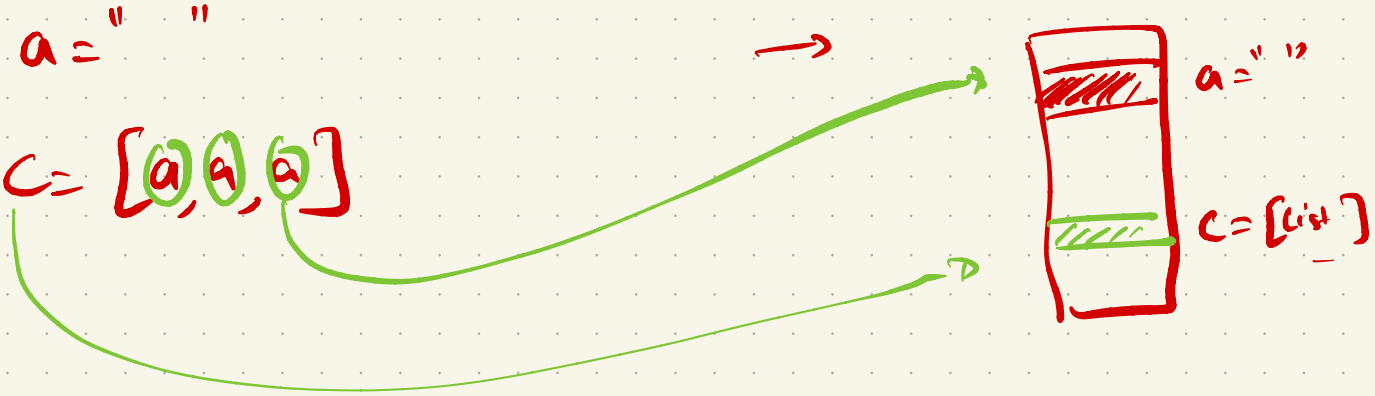
['t', 'h', 'i', 's', ...]

Python Memory →

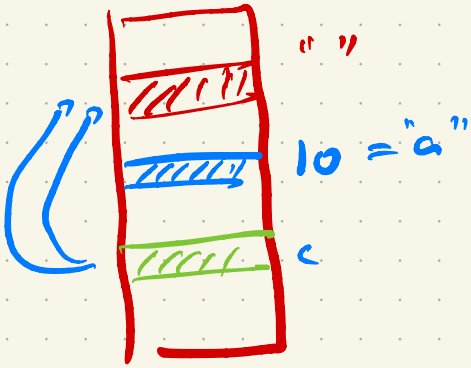
```
b = [ [" "] x3 ] x3

b --> [" ", " ", " " ]
  --> [              ]
  --> [              ]
```

a = " "

c = [a, a, a]



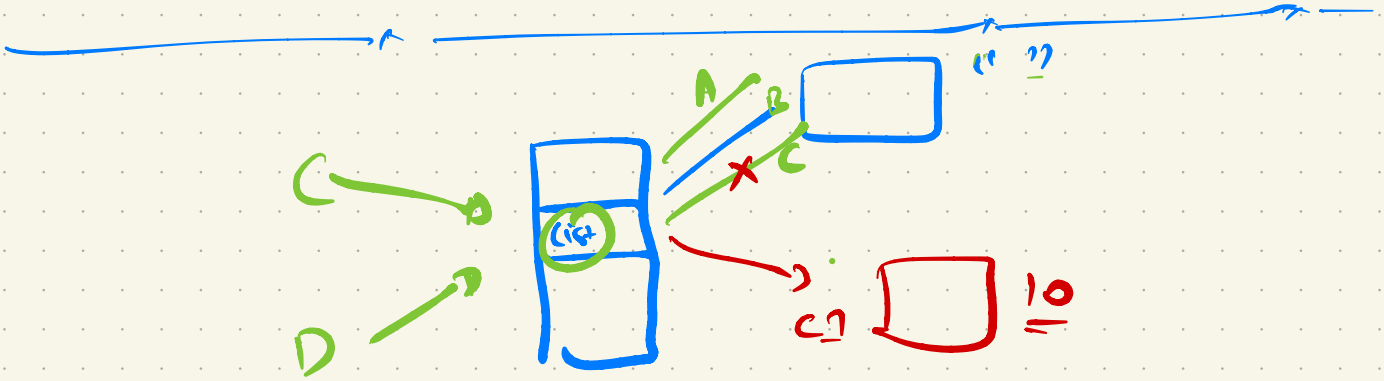
a = 10



c[0] refers to a String (immutable)

if we change c[0] only c[0] get changed

c[1], c[2] are not effected.



⇒ All mutable objects → They keep reference of memory → Further references of elements

immutable obje → final vals

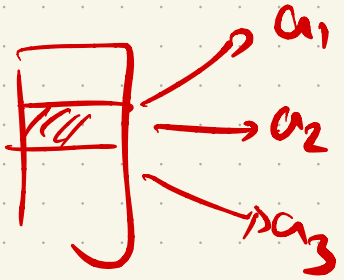
a=5

b="string"

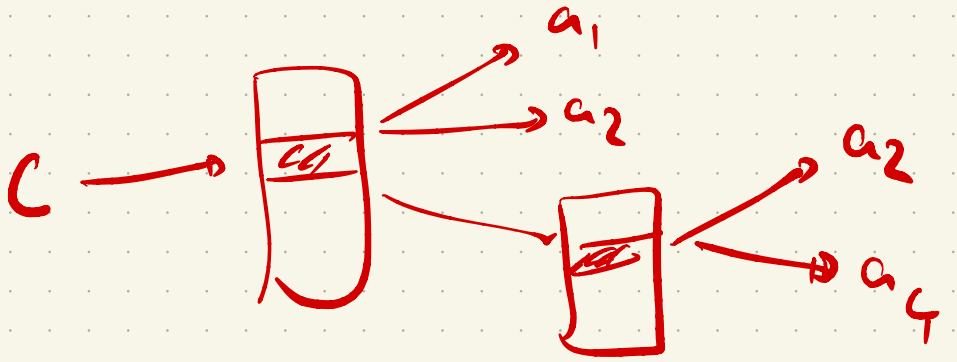
a →

c=[a1, a2, a3]

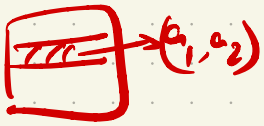
c →



c=[a1, a2, [a2, a4]]



$d = (a_1, a_2)$ tuple



d

Time Complexity → Related to algorithm analysis.

Algorithm Analysis

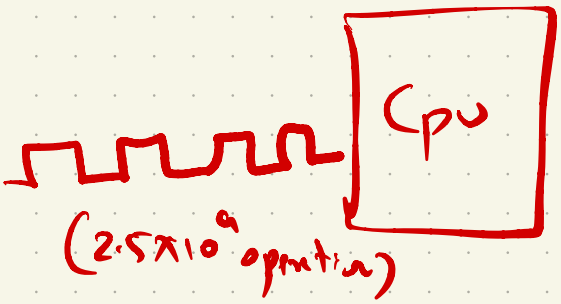
Space (Memory) [Space Complexity]

Processing (CPU) [Time Complexity]

⇒ 2.5 GHz

The system can perform 2.5×10^9 operations/sec (Clock speed)

Add 4 & 5



Time Complexity → xx How much time the algo will take ?)

↳ How many operation a algo will do.

1.5 GHz $\frac{3}{1.5 \times 10^9}$ 4+5 → 3 operations

4.5 GHz $\frac{3}{4.5 \times 10^9}$ sec

⇒ Actual Query time / Calc time is dependent on machine.

Time Complexity → number of operations

Time Complexity →

Define a Cost function | func that calculate no. of steps.

whatever is your relationship b/w n & no. of steps for that function.

$$C(n) = k_1 n^2 + k_2 n + k_3 (1)$$

Highest power is considered to be the complexity → Big O notation

$O(n^2)$

$$n^3 + n^2 + n + 1$$

Only biggest power dictates no. of steps for large n's.

$O(n^3 + n^2 + n + 1) \simeq O(n^3)$

$O(2n^3) \simeq O(n^3)$

$(2n^3) > (n^3) >>> n^2$

$2n^3 \simeq n^3$

$100n^3$ vs n^3

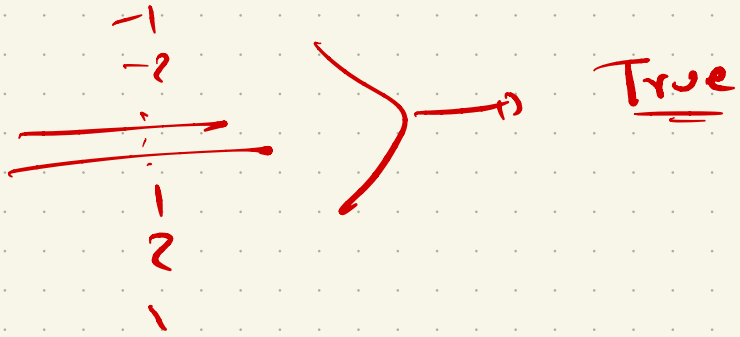
lower the Complexity the better is the performance

$n^3 >> n^2$

$n^3 \simeq n^3 + n^2$ ←

⇒ Integer/Number ⟶ Boolean

0 ⟶ False



⇒ Boolean ⟶ Integer

False ⟶ 0

True ⟶ 1

Why Complexities are important

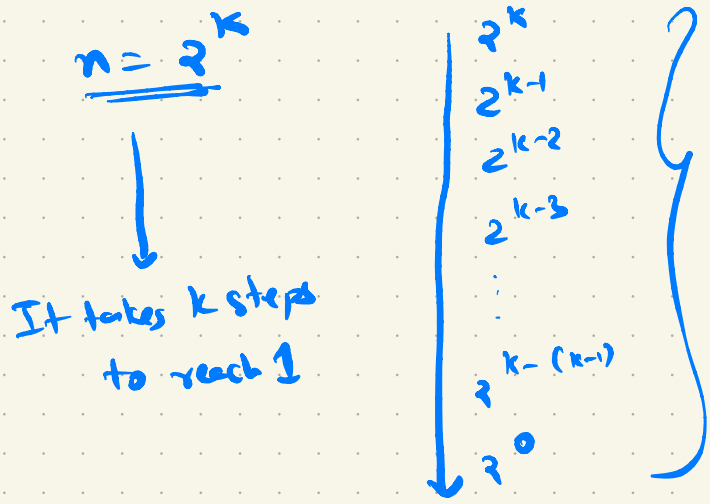
① Compute Resource are limited

→ Search
→ Recommedation } we want fast
→

Understand how much operation/ cost your algo will take

Machines → Repeatative Task →

If $n = 2^k$



for a while loop which reduce n to half every time

$n = 2^k \longrightarrow k$ steps

Complexity — k steps

$n = 2^k \Rightarrow$ **$k = \log_2 N$**

Complexity $\Rightarrow$ $(\log N)$

---

for / for → $n \to n^2 \to n^3 \to \cdots$

$\Rightarrow$ while $n//2$ $\quad n//3$ $\quad n//5$ $\quad n//10$

$n = 2^k$ $\quad n = 3^k$ $\quad n = 5^k$

$\log_2(n)$ $\quad \log_3(\cdot)$ $\quad \log_5$

$n \to n//2$ $\quad n \to n//3$ $\quad n \to n//4$

```
for i in range(n):
  j=i
  while j//2>=1:
    print(j)
```

→ log j time

$\Rightarrow \log(n!)$

$\log 1 + \log 2 + \log 3 + \cdots + \log n = \log(1 \cdot 2 \cdots n) = \log(n!)$

$\Rightarrow \log n + \log n + \cdots + \log(n) \simeq n \log n$

$\Rightarrow$ which is a better algorithm

$n \log n$ $\quad$ $\log(n!)$

$\log(n!)$ is a better algo

as $\log(n!) << n \log n$

$\Rightarrow$ Understand TC for Sorting Algorithms

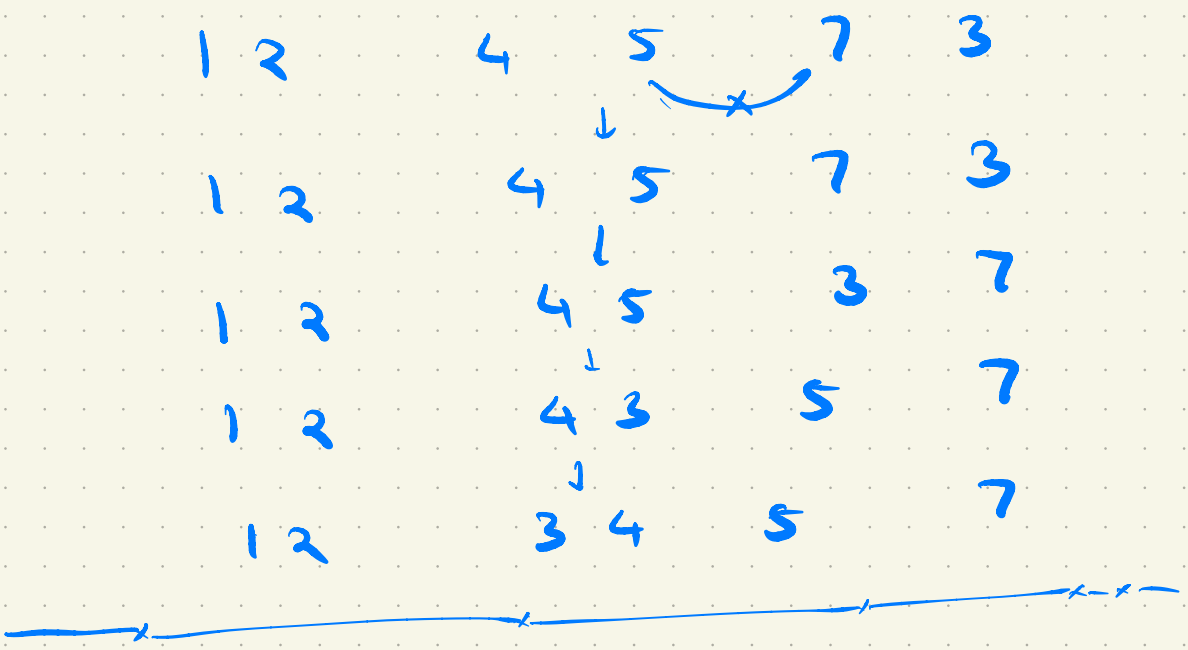
- Bubble
- Selection
- Insertion Sort
- Merge Sort
- Tim Sort

$\Rightarrow$ 5 1 2 4 7 3

Compare Swap $\quad a[i] > a[i+1] \Rightarrow$ Swap

1 5 2 4 7 3

1 2 5 4 7 3



[1,2,3,4,5,7] →

How many swaps = ?? 0

[1,2,3,4,7,5] → swaps → 1

[2,1,3,4,7,5] → swaps → 2

[7,5,4,3,2,1] → swaps →

5 times 5 4 3 2 1 7 — 5+4+3+2+1

4 times 4 3 2 1 5 7

3 times 3 2 1 4 5 7

2 times 2 1 3 4 5 7

1 time 1 2 3 4 5 7

15 swaps

⇒ Bubble Sort →

Best Case → 0 swaps

Worst Case →

n-1, n-2, ... 1 ⇒ 1+2+3+... + n-1 swaps

1+2+... +(n-1) $\frac{n^2-n}{2}$ swaps

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

0 ⟷ $\frac{(n)(n-1)}{2}$

⇒ Worst Case Complexity $\frac{(n)(n-1)}{2}$ ← $O(n^2)$

If data is provided → find no. of steps → then give Complexity

Bubble Sort → Best Case → 0 Swaps $O(n)$

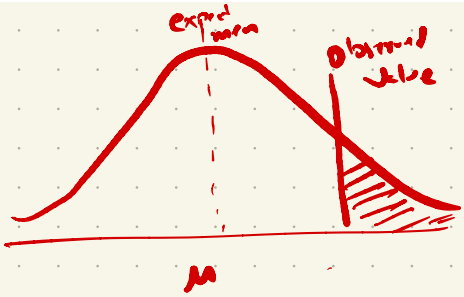
atleast 1 Complete iteration of check would be needed.

1, 2, 3, 4, 5, 6 Comparison → $n-1$ (Con

A fitness App claims that its users walk an average of 8,000 steps per day. A random sample of 30 users showed an average of 7,600 steps per day with a standard deviation of 1,200 steps. Conduct a right-tailed Z-test at a 5% significance level to determine if the App's claim is supported. What is the p-value

Right tailed test → Values are on right side of median.

↳



How much chance / probability that data will lie on right side of observed value

[30 sample → 7600
→ 8000
→ 8500
→ 8900] distribution of Tyre

As a data analyst at Global Mariotech, one of your prime duties is to understand customer sentiments and provide a better experience to the customers.
Based on a recent survey, it was found that there were 30% negative responses, and 70% positive.Assuming that we pick one response after another randomly, estimate the number of feedbacks to expect, before a negative feedback appears....solve using geometric distribution..

0.7 → P 0.3 → N

→ N N N N N N N P
(0.3)(0.1)(0.3) ... (0.7) → $(0.3)^7(0.7)$ → 8

→ N N N P → $(0.3)^3(0.7)$ → 4

→ P → (0.7) → 1

1M

0.7

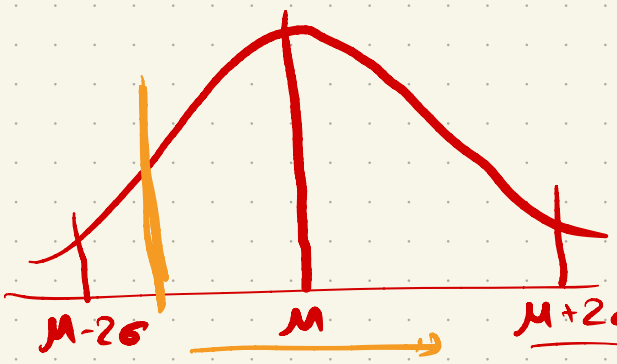
0.21

(0.7)	→ 1
(0.3)(0.7)	→ 2
$(0.3)^2(0.7)$	→ 3

$E(x) = (0.7) + (0.21) \times 2 + \dots$

(0.7) + (2)( )( )
+(3)( )( )
+ ...

1M



95% chance

⇒ I just randomly get a sample → 30 size

μ_{sample} σ_{sample}

Sample mean

Std dev

$\frac{\sigma_{pop}}{\sqrt{n}}$

95% → any random SM will lies b/w $\mu_{pop} \pm \frac{2\sigma_{pop}}{\sqrt{n}}$

$\mu_{SM} \cong$

Collab : https://colab.research.google.com/drive/1tWblCOWJ_CmTpsZtSafT3MyTLbWchAp4?usp=sharing