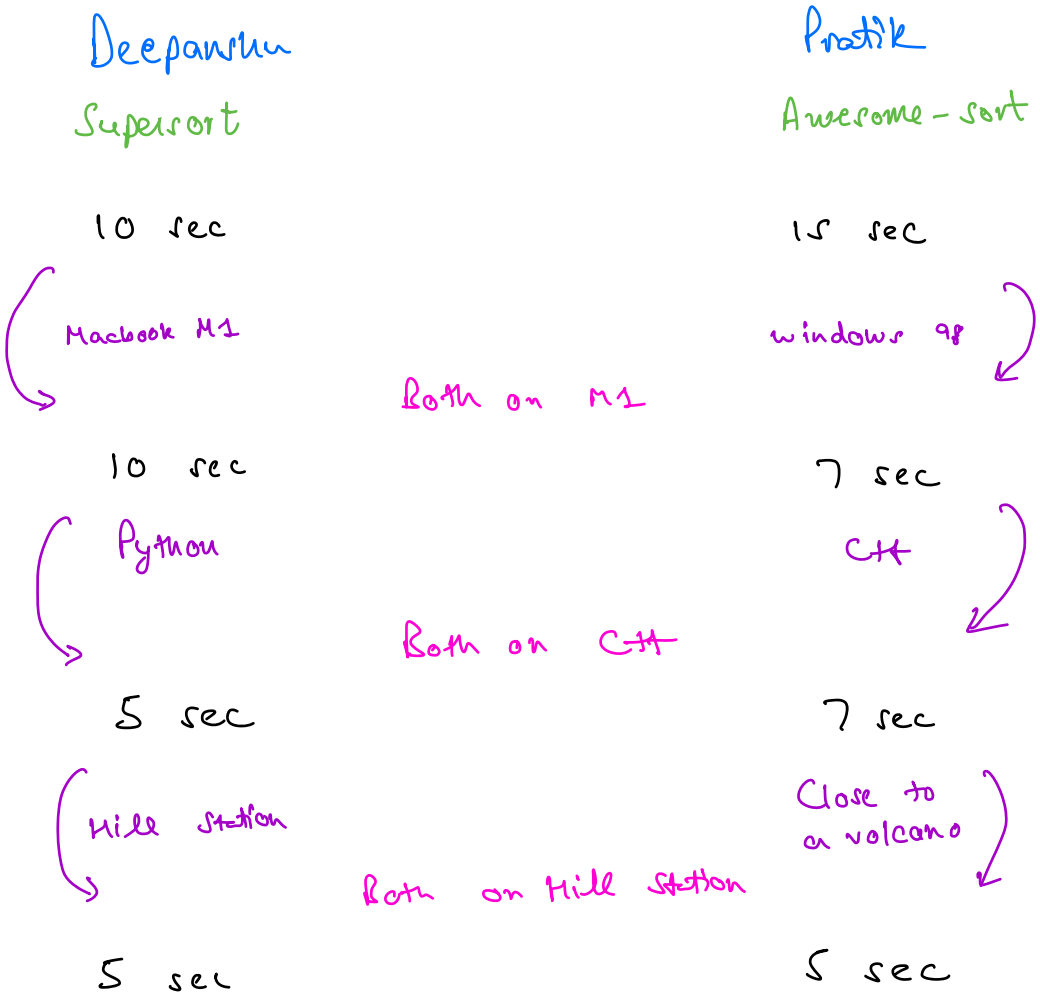
Time Complexity 2

AGENDA:

- Asymptotic Analysis
- Big O
- Issues with Big O
- Space complexity
- TLE

Q. Sort the array.



Same

Execution Time - Not a reliable measure of performance

→ It depends on a lot of factors

- → Software
- → Hardware
- → Environment

```
for i in range(n):
```

→ N iterations

No of iterations is 'independent' of all external factors

Asymptotic Analysis of Algorithms

Observing performance of algorithms for very large inputs.

Big O

Comparisons using Big O Notation

$5 \log_2(N)$ → $O(\log N)$ Better

$10 N^2$ → $O(N^2)$

→ Calculate no of iterations

→ Neglect all lower order terms

→ Neglect constant coefficient

← Independent of external factors

Why neglect lower order terms ?

Algo: $N^2 + 10N$ ← lower order term

$N = 100 \quad = \quad 10^4 + 10^3$

% contribution of lower order term $= \frac{10^3}{10^2 + 10^4} \times 100 = 9.09\%$

$N = 10^4 \quad = \quad 10^8 + 10^5$

% contribution of lower order term $= \frac{10^5}{10^5 + 10^8} \times 100 = 0.1\%$

Contribution of lower order terms is negligible / significantly smaller for larger inputs.

Why neglect constant coefficient ?

Arun	Nithin
$10 \log N$	N
$10^2 \log N$	N
$10^3 \log N$	N
$10^4 \log N$	N
$10^4 \log N$	$10N + 10^6$

Issues with Big-O

	Anand	Sreekanth	More efficient
	$100N$	N^2	
$N=50$	100×50	50×50	Sreekanth
$N=80$	100×80	80×80	Sreekanth
$N=100$	100×100	100×100	Same
$N=120$	100×120	120×120	Anand
$N=150$	100×150	150×150	Anand
	$O(N)$	$O(N^2)$	

Issue 2

Rashmi

$N^2 + 10N$

↓

$O(N^2)$

Tavish

$2N^2 + 5N$

↓

$O(N^2)$

Issue: Big O says both are same

$N^2 + 10N$

↑

More efficient

$2N^2 + 5N$

Big O will solve 99% of your problems

Linear Search

arr = [5, 2, 8, 1, 6 ....]

key = 50

```
for i in range( len(arr)) :
    if  arr[i] == key :
        return True
return false
```

Best - O(1)

Worst - O(N)

Default - Worst

Manager

Best - 1 day

Worst - 3 months

Hope for the best

Prepare for the worst

Code → Time Complexity

→ Space Complexity

Space Complexity

Amount of extra space taken by your algorithm.

Q1

```
func (int N) {
    int x = N
    int y = x + x
    int z = x + y
}
```

int - 4 bytes

3 x 4 = 12 bytes

↓

O(1)

Q2

int - 4 bytes

```
func (int N) {
    int x = N          - 4B
    int y = x^2        - 4B
    int z = x+y        - 4B
    arr[] = int [N]    - 4N bytes
}
```

`arr[]` = int `[N]` ← Array of integers

`N` ↑ Size of array

$4 + 4 + 4 + 4N$

$= (12 + 4N)$ bytes

↓

$O(N)$

Q3

$4 + 4 + 4 + 4N + 4N^2$

↓

$O(N^2)$

```
func (int N) {
    int x = N          - 4B
    int y = x^2        - 4B
    int z = x+y        - 4B
    arr[] = int[N]     - 4N bytes   ← 1D Array of size N
    arr = [0] * N
    a[][] = int[N][N]  ← 2D Array of N×N
                       ↓
                       4N^2 bytes
}
```

Given N elements, calculate their sum.

```
                Array of int        Size of array
                    ↓                   ↓
sumOfArray ( int  arr[], int N ) {

    sum = 0        - 4B
                  ↗
    for ( i = 0 ; i < N ; i++ ) {

        sum = sum + arr[i]

    }
    return sum
}
```

4 + 4 = 8 B

↓

O(1)

Aditya	Antara
$100 \log N$	$N/10$

Before 3550 : Antara wins

After 3550 : Aditya wins

Overall Aditya wins

Hotstar - 2×10^9 streamed

Google - 10^9 results searched in 0.123...s

Despacito - 7 B views

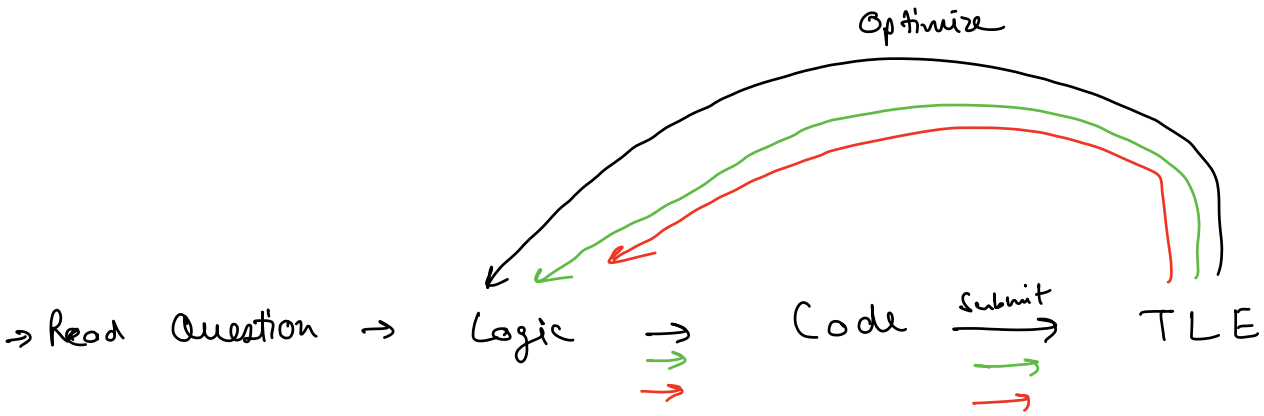
Huge data

Data is increasing

TLE - Time Limit Exceeded

Google Contest – 2 Questions

1 hr



// Check whether TLE

without even writing a single line

of code

Online Judge → Servers → 1 GHz

↓

Time Limit - 1 sec

↓

10^9 operations / second

instructions

At max we can have

10^9 operations / instructions

↓

Variable declaration

+, -, ÷, *

function call

```
i = 0                    ← once
while i < n:
    if i % 2 == 0:
        print(i)
    i += 1
```

Instructions:

1 + 5N

Assumption

1 iteration $\longrightarrow$ 10 instructions

At max we can have 10^9 instructions

10^8 iterations $\rightarrow$ 10^9 instructions

At max we can have 10^8 iterations (Approx)

Process to solve

1) Read & understand the question

2) Logic

3) Correctness

Try for multiple testcases

4) Check if TLE occurs

If TLE

↓ If no TLE

5) Write the code

6) Submit

Ex 1

Ex 2

Ex 3 ... 5 → 6, 7, 8, 9, 10

↳ logic →

Constraints

Eg - $1 <= N <= 10^5$

$1 <= arr[i] <= 10^8$

Algo - $O(N^2)$ time

$N = 10^5 \rightarrow 10^{10}$ iterations

$\Rightarrow$ TLE

Eg -

$1 <= N <= 5 \times 10^3$

Algo - $O(N^2)$ ← Not optimized

$N = 5 \times 10^3$

$N^2 \rightarrow (5 \times 10^3)^2 \rightarrow 25 \times 10^6$ iterations

$< 10^8$

Will work

No TLE

Eg

$1 <= N <= 5 \times 10^2$

Algo : $O(N^3)$

Rare Case

$N = 5 \times 10^2$

$N^3 \rightarrow (5 \times 10^2)^3 \rightarrow 125 \times 10^6$

$= 1.25 \times 10^8$ iterations

Can't be said

If you have a code $\leq 10^8$ iterations

Corner Case

Practice

Every Problem → Time & Space

Doubts

++i
i++
} Not in Python

i = i+1

Thank You

i = 5

print (i++) → 5

print (i) → 6

Doesn't work in Python

---

i = 5

print (++i) → 6

print (i) → 6

Doesn't work in Python

---

i = 5

print (i = i+1) → Error

print (i += 1) → Error


Good
Night


Thank
You


Wednesday