friday

Notes

Topic Name : Time - Complexity 2

TODAY: => 1 - Comparing 2 Algorithms

=> 2 - Asymptotic Analysis : Big-O

Why neglect lower order terms?

Why drop constant coefficient?

AGENDA:

3 - Issues in Big-O notation

4 - Time & Space Complexities

5 - Mathematical and Graphical Representation of Big-O. Problems.

6 - Why TLE? How to know before submitting code.

7 - Worst Case, Best Case, Avg Case

Lists

Lists Arrays

Big-O Notation

How to find Big-O Notation :

1. Calculate the no. of iterations based on the Input size.

2. Neglect the lower order terms.

N^2 , N , N^3

$N^3 + N^2 + N$

$= O(N^3)$

Keep the highest order term.

3. Neglect the constant coefficients.

$4N^2$ $\quad$ $6N^2$

$= O(N^2)$ $\quad$ $= O(N^3)$

Order of N^2

Big-O

Quiz-1

$F(N) = N^3 + 10N^2 + 8$

$8 = 8 \cdot N^0$

$N^3 + N^2 + N^0$

$F(N) = C \cdot N^0 = O(1)$

C is constant.

Mathematical Relation : Common Complexities.

$N > 2$

$$\log(N) < \sqrt{N} < N < N\log N < N\sqrt{N} < N^2 < 2^N$$

$N = 64$

$\log_2[N] = \log_2 64 = 6$

$\sqrt{N} = \sqrt{64} = 8$

$N = 64$

$N \cdot \log(N) = 64 \cdot 6 = 384$

$N\sqrt{N} = 64 \cdot 8 = 512$

$N^2 = 64^2 = 4096$

$2^N = 2^{64}$

Special Names

- $O(1)$ = constant
- $O(N)$ = Linear
- $O(N^2)$ = Quadratic
- $O(N \log N)$ = Linear Logarithmic
- $O(\log N)$ = logarithmic
- $O(2^N)$ = Exponential

N^t $t \geq 2$ → Polynomial ($N^2, N^3, N^4, N^5 \ldots$)

N^3 → Cubic

Quiz-2

$$F(N) = N\log(N) + 7N^2 + 8$$

$$= O(N^2)$$

Time & Space => Resources.

Mohit Sharma

Swathi Yadav

Task: Given list containing N integers, sort them in ascending der.

Inp: [1, 7, −1, 2, 4, 6] Same Input. [100, 2, - - - - 1]

100 element.

out: [−1, 1, 2, 4, 6, 7]

(1) Tuktuk sorting Algo (TSA)

(2) Ascending Algo (AA)

Execution Time:

i) Windows 95 — 15 sec.

Macbook Pro M1 — 10 sec.

Dependency

a) OS

ii) 7 sec.

C++

Python.

10 sec.

Python

b) Programming Lag.

C++ is faster than Python.

iii) 12 sec.

Top of a volcano.

10 sec.

Canada.

c) Hardware

Processor

Temperature

(External)

iv) 5 sec.

10 sec

Observation:

⇒ Execution Time depends on a lot of external factors.

↓ Not a good measure of comparing time complexity. (TC)

(Q) What is a good comparison measure for TC

Iterations is an independent factor.

⇒ $T(N) \propto I(N)$

Time taken for input of size N is directly proportional to the no. of iterations for input N.

$I(N) = N$

If N↑ the no of iterations increases so does the time taken by the program.

Mohit	Swathi
TSA	AA
No. of iterations: $100 \log(N)$	$N/10 = \frac{1}{10} * N$
$= O(\log N)$	$O(N)$

$\Rightarrow$ We will test their algorithm on diff. values of N.

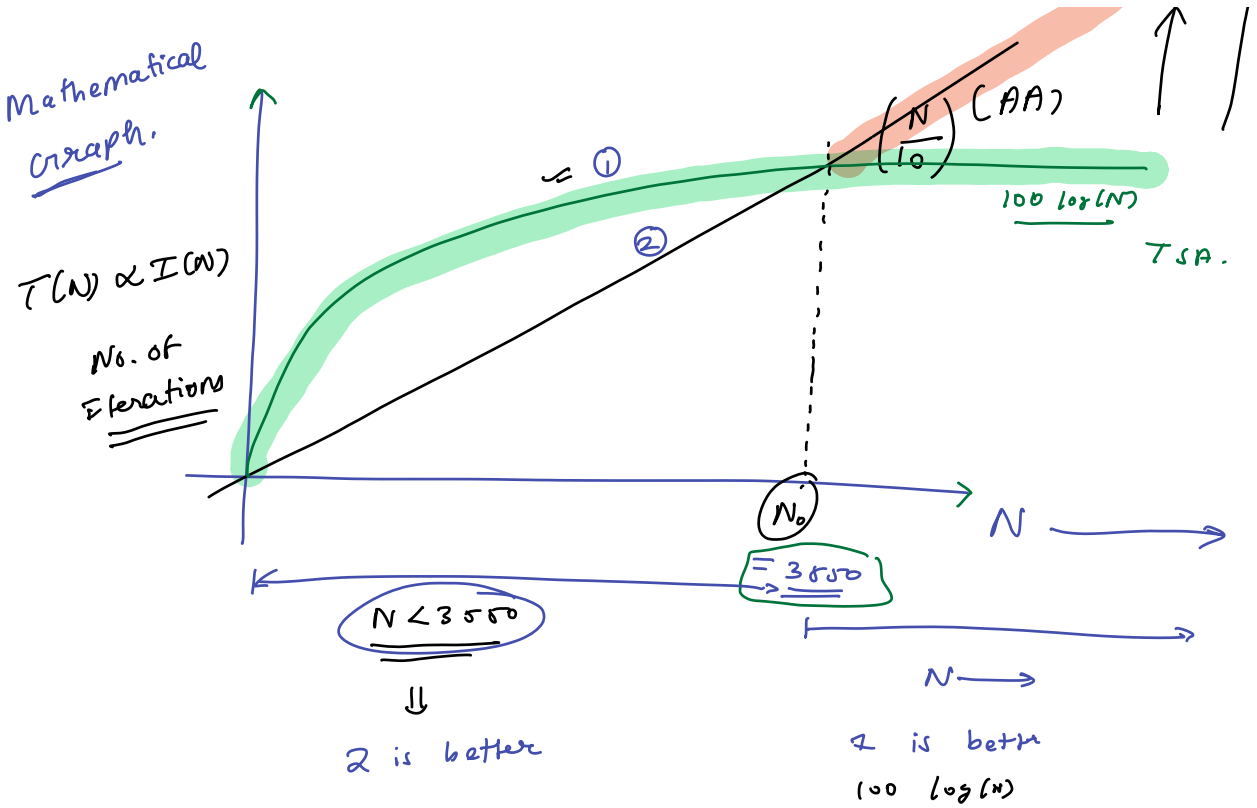
In general, N values will be very HUGE.

As $N \to \infty$

$\Rightarrow$ Asymptotic Analysis $\Rightarrow$ Finding TC as N tends to infinity.

$\Downarrow$

Big-O

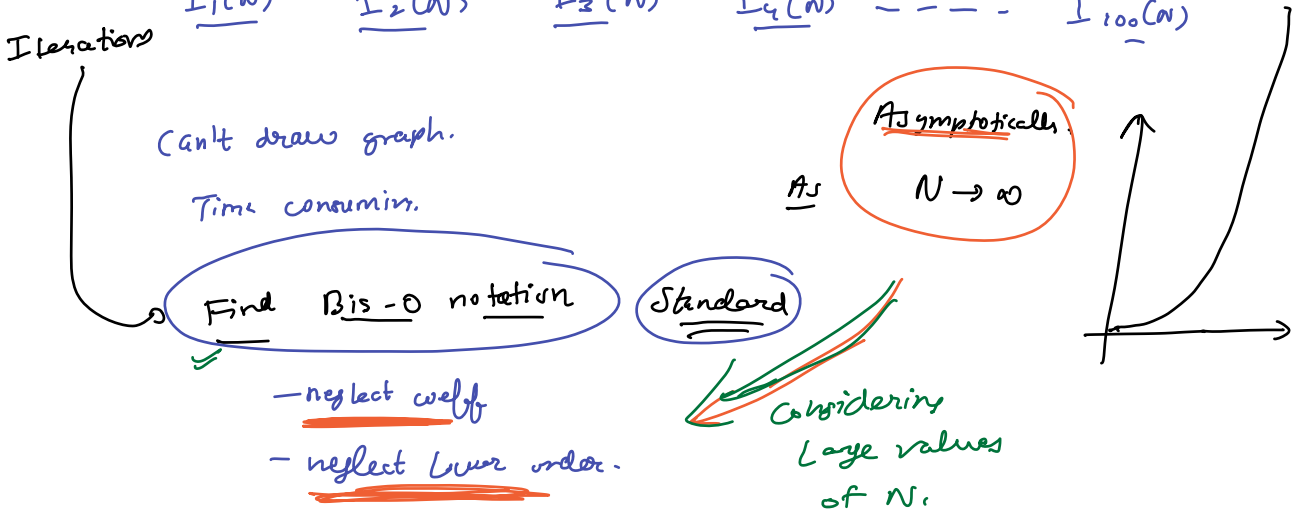


As per Asymptotic Analysis, we will prefer 1.

1 takes less time as $N \to \infty$.

(Q) Everyone in the class comes up with some sorting algo.

$I_1(N)$ $I_2(N)$ $I_3(N)$ $I_4(N)$ ---- $I_{100}(N)$



$I_1(N) = N^2 + N\log N + N\sqrt{N} + N^3 + 4N\sqrt{N} = O(N^3)$ ✓

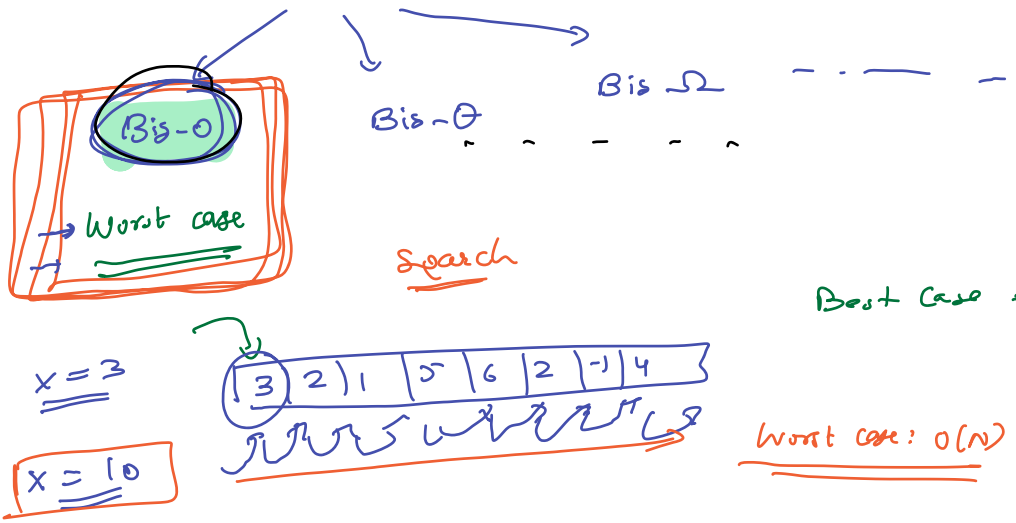
$I_2(N) = N^2\sqrt{N} + N^4 + N^2\log N\sqrt{N} + N^2 \leq O(N^4)$

$I_3(N) = N^2 + N$

$= O(N^2)$ ✓

Performance of Algo as N becomes very large.

Asymptotic Analysis



Searching for x in list.

```
def search(l, x):
    for(ele in l):
        if(ele==x):
            return ele
```

Next class.

List size of N

[⟷]

N elements.

Screen - 1

No. of iterations = ? 1 or 2 or 3 - -- or N

N, X

Best Case

Worst Case

Quiz

$f(N)$ $N^2 + 10N + 8$

$= O(N^2)$ ⇒

$= O(N^3)$?

$= O(N^4)$

Mathematical definition

$f(N)$ is $O(g(n))$ if

there exist 2 constants c, n_0

such that

$f(N) \leq c \cdot g(n)$ for all $n \geq n_0$

✓

Observation: $c \cdot g(n)$ is an upper bound for $f(n)$

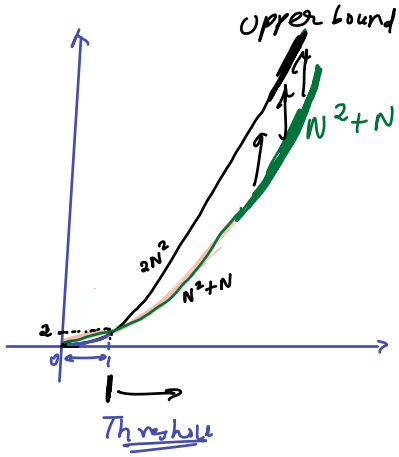
Es

$f(N) = N^2 + N = O(N^2)$

$g(N) = N^2$

$c = 2$

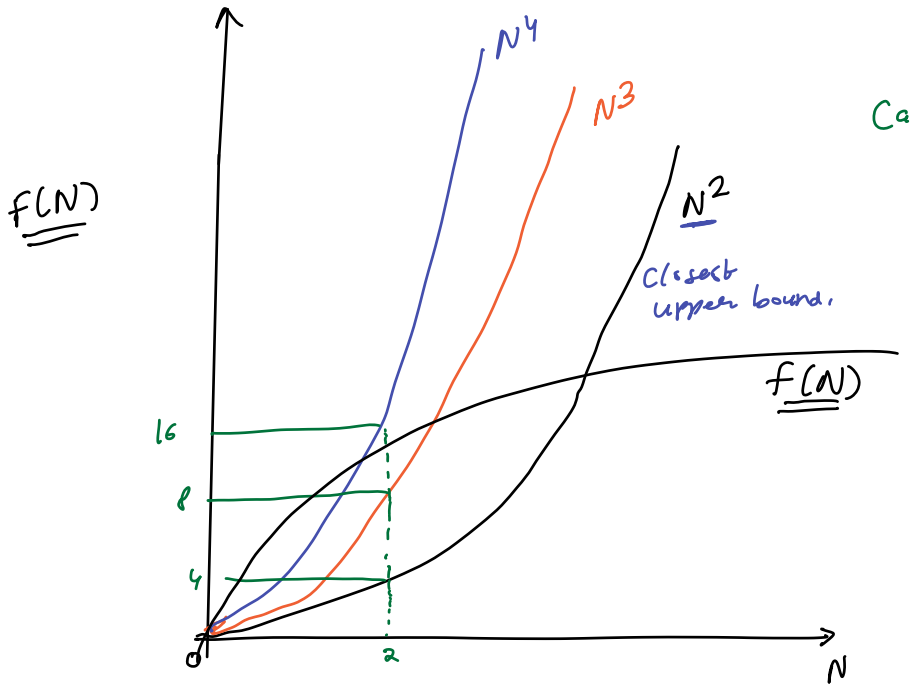
$N^2 + N \leq 2N^2$



After $n_0 = 1$

$2N^2$ is always larger than N^2+N

	N^2+N	$2N^2$
$N=0$	0	0
$N=\frac{1}{2}$	$\frac{1}{4}+\frac{1}{2} = \frac{3}{4} = 0.75$	$2 * \frac{1}{2} * \frac{1}{2} = \frac{1}{2} = 0.5$
$N=1$	2	2
$N=2$	$4+2=6$	$8*4=8$
$N=5$	$25+5=30$	$2*25=50$



Can I say N^3 & N^4 are both upper bounds for N^2?

$f(N)$ is $O(g(n))$ if

there exist 2 constants c, n_0

such that

$f(N) \leq c \cdot g(n)$ for all $n \geq n_0$

$\forall$

$f(N) = N^2 + 10N + 8$ is $O(N^3)$

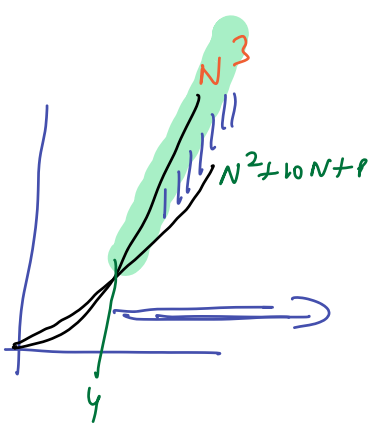
$g(N) = N^3$

$c = 1$

$n_0 = 4$

$N^2 + 10N + 8 <= N^3$

$16 + 40 + 8 = 64$ $\quad$ 64



$\Rightarrow f(N) = N^2 + 10N + 8$ is $O(N^2)$

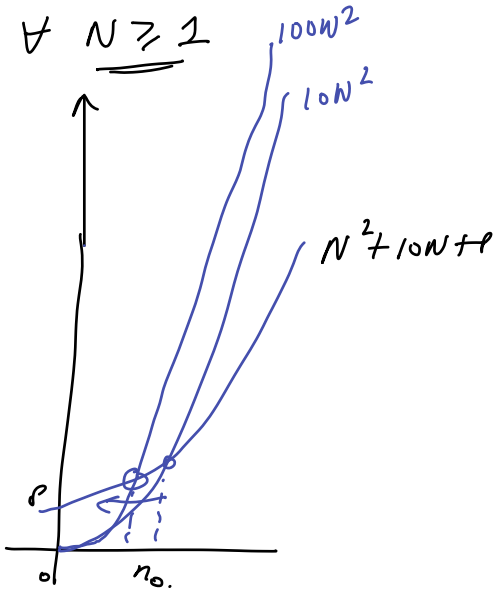
$c = 100$ → $c = 100000$

$N^2 + 10N + 8 \leq 100 N^2$ $\quad \forall\ N \geq 1$

N	$N^2 + 10N + 8$	$100 N^2$
$N = 1$	$1 + 10 + 8 = 19$	100

$n_0 = 1$

$f(N)$ is $O(N^2)$



✓ → Drop lower order terms

✓ → Drop constant coefficients

✓ $f(N) = N^2 + 4N + 8 = O(N^2)$

⟹ Mathematical Definition

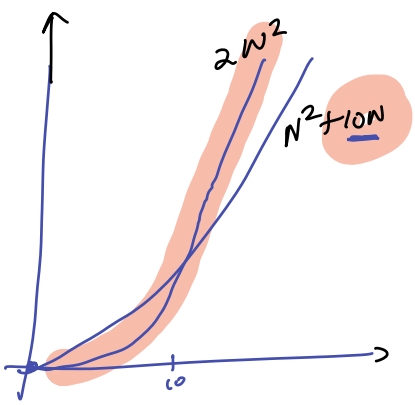
independent of constants & lower order terms.

$f(N)$ is $O(g(n))$ if there exist 2 constants c, n_0 such that

$f(N) \leq c \cdot g(n)$ for all $n \geq n_0$ ✓

$C=2$

$N^2 + 10N <= 2N^2$



$N^2 + 10N = 2N^2$

$\Rightarrow 2N^2 - N^2 = 10N$

$\Rightarrow N^2 - 10N = 0$

$\Rightarrow N(N-10) = 0$

$\Rightarrow N = 10$

strictest closest upper bound

$N^2 + 10N = O(N^2)$

$O(N^3)$

$O(N^4)$

Car

Bicycle

Motorbike

Truck

group similar type.

Speed of Truck is less than Speed of light.

Comparing 2 algorithms

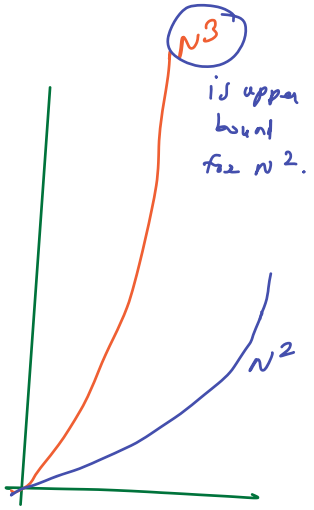
1) Find Big-O.

2) One which has highest order term is not efficient as compared with others.

TSA	AA
$O(\log N)$	$O(N)$

Quiz

1. $F(N) = n^3$ is $O(N^2)$ ✗
2. $F(N) = N^3$ is $O(N^4)$ ✓



$N^3 < c \cdot N^2 \quad \forall \quad n \geq n_0$

? ?

This will dominate after a certain threshold.

$N^3 <= N^4 \quad \forall \; n \geq 1$

$c = 1, \; n_0 = 1.$

Doubts

Worst Case

```
if (  ) {
    f()
} else() {
    g()
}
```

max(f, g)

max(N, N^2)

$= N^2$

Find Big-O for this.

$f(n) <= c \cdot g(n)$

$N^2 + 10N$

$= O(N^2)$

g(n)

1. dropping lower order
2. dropping constant coefficients.

HW - Q. 1

1) Try to contact TA.
2) Message on Slack.

Extra Reference Session