

Time Complexity - II

Today's Content

- Comparing two algos
 - Using execution time
 - Using iterations & graphs
- Why Big O needed?
 - Why lower order terms are neglected
 - Why const. coefficients terms are neglected
 - Issues in Big O
 - Worst Case
- Space Complexity
- TLE
 - Why TLE occurs?
 - How to approach any given problem
 - Importance of constraints

Comparing Algos using Execution Time

Given N elements, sort them in increasing order.

$N = 10^4$ (Input size)

Algo 1 {Azad}

Algo 2 {Aniket}

Exec. Time

15 sec

10 sec

↓

↓

(Windows XP)

(Macbook M₂)

↓

|

(Macbook M2)

↓
7 sec

↓
(C++)

↓
7 sec

↓
(Top of hot volcano)

↓
Mt. Everest

↓
5 sec

↓
10 sec

↓
(Python)

↓
(C++)

5 sec

Mt. Everest

↓

5 sec

Execution time : It depends on so many external factors,
hence we generally don't compare
exec. time b/w 2 algs.

Comparing Using Iterations & Graphs

Algo1 { Sathwik }

Algo2 { Jamal }

iterations

$100 \log_2(N)$

$N/10$

for $N=10$

$100 \log_2(10) \rightarrow \sim 300$

$10/10 \rightarrow 1$

for $N \leq 3900$

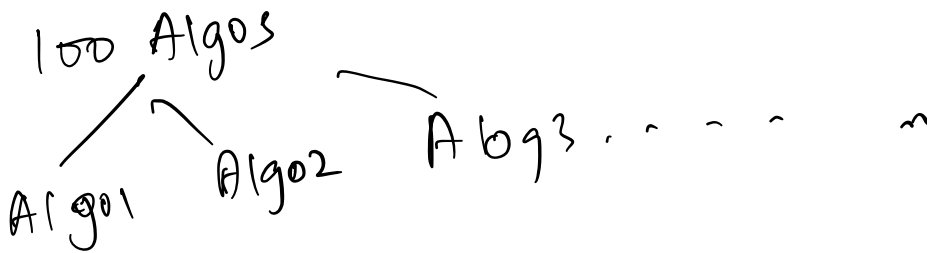
$$N > 390$$

Age 2 is better

Argo 1 is better

Google results: 1M+

Baby Shark: 10.84 Billion



Asymptotic Analysis of Algorithms

↳ Performance analysis of Algos for very large inputs

Use Big O notation

1. Calculate iterations based on input
2. Take higher order term & neglect lower-order }
3. Ignore const. coefficients

Why neglect lower order terms?

$$N^2 + 10N \text{ [given]}$$

Input size

$$N \geq 10$$

Total iterations

200

% of lower order terms

$$\frac{10\text{N}}{20\text{N}} \cdot \frac{100}{200} = 50\%$$

$10^3 \sim 9\%$

$$N = 100$$

$$N = 10^3$$

$$N = 10^6$$

$$10^4 + 10^3$$

$$10^6 + 10^4$$

$$\frac{10^4}{10^6 + 10^4} = < 1\%$$

Insignificant

Why ignore const. coefficients.

$$10N$$

$$N = 10^5$$

$$N$$

$$10^6$$

$$10^5$$

Claim 1: for all inputs, we can decide which Algo is better. ✗

Claim 2: for all inputs $\geq x$, we can decide which Algo is better. ✓

Final Claim: When we compare two Algos using Big O, Algo1 will always be better than Algo2 for all inp. values above a certain point.

↳ threshold

* After threshold, Big O holds.

* Please don't worry about threshold

Issues in Big O

$$2N^2 + 4N$$

$$O(N^2)$$

$$2N^2 - 4N - 2N^2$$

$4N$ is better than N^2

$$3N^2$$

$$O(N^2)$$

$$3N^2 - 2N^2$$

→ If we have same Big O for 2 algos, then Big O will fail.

Worse Case

Ques Searching of an element $= K$

bool search (int a[], int K) {

for (int i=0; i < a.size(); ++i)

if (a[i] == K)

return true;

}

return false;

}

total iteration

$$= N$$

$$O(N)$$

best-case iteration = 1
worst-case iteration = N

Munger $\rightarrow \{ \text{Task} \}$
5 days \swarrow Best case
30 days \searrow Worst case

Break: 10:11 — 10:21 PM

Space Complexity

Code \rightarrow Time Complexity
Code \rightarrow Space Complexity

2GB more space

my code will take 5GB more ?

```

void func (N) {
    int x = 0;
    int y = N;
    long p = 5;
}

```

$\text{int} \rightarrow 4B$
 $\text{long} \rightarrow 8B$
 $16B$

3

```

void func(int a[]) {
    int m = a[0];
    for (i = 1; i < n; ++i) {
        m = max(m, a[i]);
    }
}

```

3

return m;

3

$4N$
 4
 8
 $4N + 4$
 Space $O(1)$

```

void func(int a[], int n) {
    int pf[n];
    pf[0] = a[0];
}

```

$\rightarrow 4 \times n B$
 $\rightarrow 4B$

```

for (i = 1; i < n; ++i) {
    arr[i] = arr[i] + arr[i-1];
}

```

}

$O(1)$ X

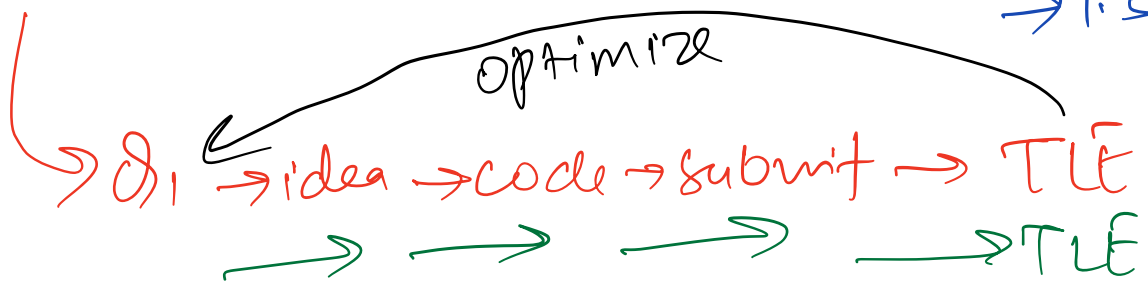
}

Space Complexity = $O(N)$ ✓

$4n + 4B$

Time limit Exceeded TLE

Bhaurajet → (Google) → HC → 3Q
→ 1.5hr



Online Editors → 1 GHz → 10^9 instructions
1 sec

$s = 0; \quad +1$

$\text{for}(\text{int } i = 0; \quad +1 \quad i < n; \quad +1 \quad ++i) \{$

$\quad s = s + i; \quad +1$

iteration = N

3

total instruction =
 $2 + 3n$

Approx! :

1 instruction = 1 iterations

Approx!!

1 iteration = 100 instructions

10^9 ins \rightarrow 1 sec

1 sec \rightarrow $[10^7 - 10^8]$ iterations

Importance of Constraints

$$1 \leq N \leq 10^6$$

Algo $\rightarrow O(N^2)$ ✗
iteration $\stackrel{N=10^6}{=} \underline{\underline{10^{12}}}$

$O(N \log N)$ ✓✓

$\hookrightarrow 10^6 \log 10^6$

$\hookrightarrow 10^6 \times 19$

$\approx 1.9 \times 10^7$

$$N = 10^4$$

$$O(N^2)$$

$\Rightarrow 10^8$ iterations

$$1 \leq N \leq 100$$

$O(N^3)$ $\xrightarrow{N \leq 100}$

10^6 iterations

no need \downarrow
 $\{O(N), O(N^2)\}$