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)

	Algo1 {Azad}	Algo2 {Aniket}
Exec. time	15 sec ↓ (Windows XP) ↓	10 sec ↓ (Macbook M_2)

(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 algos.

Comparing Using Iterations & Graphs

Algo1 {Sathwik} Algo2 {Jamal}

iterations $100\log_2(N)$ $N/10$

for $N=10$

$\boxed{100\log_2(10) \rightarrow \sim 330}$

$\boxed{10/10 \rightarrow 1}$

for $N \leq 3900$ Algo 2 is better

$N > 3900$ Algo 1 is better

Google results: 1M+

Baby Shark: 10.84 Billion

100 Algos

Algo1 Algo2 Algo3 . . . . n

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?

$N^2 + 10N$ [given]

Input size	Total iterations	% of lower order terms
$N = 10$	200	$\frac{10N}{N^2+10N}$ $\frac{100}{200} = 50\%$
	10^3	$\sim 9\%$

$N = 100$ $\quad 10^4 + 10^3 \quad \dfrac{10}{10^4+10^3}$

$N = 10^3$ $\quad 10^6 + 10^4 \quad \dfrac{10^4}{10^6+10^4} = < 1\%$

$N = 10^6$

Insignificant

Why ignore const. coeffients.

$10N \qquad N$

$N = 10^5 \qquad \underline{10^6} \rightarrow 10^5$

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

Claim 2: for all inputs $>= x$, 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

$2N^2 + 4N$ $\qquad\qquad$ $3N^2$

$O(N^2)$ $\qquad\qquad$ $O(N^2)$

$2N^2 - 4N - 2N^2$ $\qquad\qquad$ $3N^2 - 2N^2$

$4N$ is better than N^2

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

Worse Case

Qus 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

Manager → { Task }

5 days → Best case

30 days → worst case

Break: 10:11 — 10:21 PM

Space Complexity

Code → Time Complexity

Code → space complexity

2GB more space

my code will take 5sec more?

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

int → 4B
long → 8B

16B

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

4N

4

8

4N+4

Space = O(1)

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

$\rightarrow$ 4×n B

$\rightarrow$ 4B

```
for (i=1; i<n; ++i ) {
    pf[i] = a[i] + pf[i-1];
}
}
```

O(1) ✗

Space Complexity = O(N) ✓

4n + 4 B

Time Limit Exceeded TLE

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

↳ Q1 → idea → code → submit → TLE

optimize

→ → → → TLE

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

$s = 0;$ $+1$

```
for (int i=0; i<n; ++i){
    s = s + i;
}
```

$+1$ (int i=0), $+1$ (i<n), $+1$ (++i), $+1$ (s = s + i)

iteration = N

total instruction = $2 + 3n$

Approx! :

10 instructio = 1 iterations

Approx! :

1 iteration = 100 instructions

10^9 ins $\longrightarrow$ 1 sec

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

Importance of Constraints

$1 \le N \le 10^6$

Algo $\rightarrow$ $O(N^2)$ ✗

iteration $\overset{N=10^6}{=} 10^{12}$

$O(N \log N)$ ✓✓

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

$\hookrightarrow 10^6 \times 19$

$\approx 1.9 \times 10^7$

$N = 10^4$

$O(N^2)$

$\Rightarrow 10^8$ iterations

$1 \le N \le 100$

$O(N^3) \xrightarrow{N=100} \boxed{10^6 \text{ iterations}}$

$\Downarrow$

~~No~~ not need $\{O(N), O(N^2)\}$