Calculate iterations

Ranjith

Spectacular Algo

30sec

Dell windows

↓

Google

(Mac)

5sec

↓

C++

Zahara

Zee's algo

10sec

Mac M2

↓

python

↓

C++

(2sec)

1) Execution time does not determine efficiency of an algorithm

why?

Execution time depends on external factors

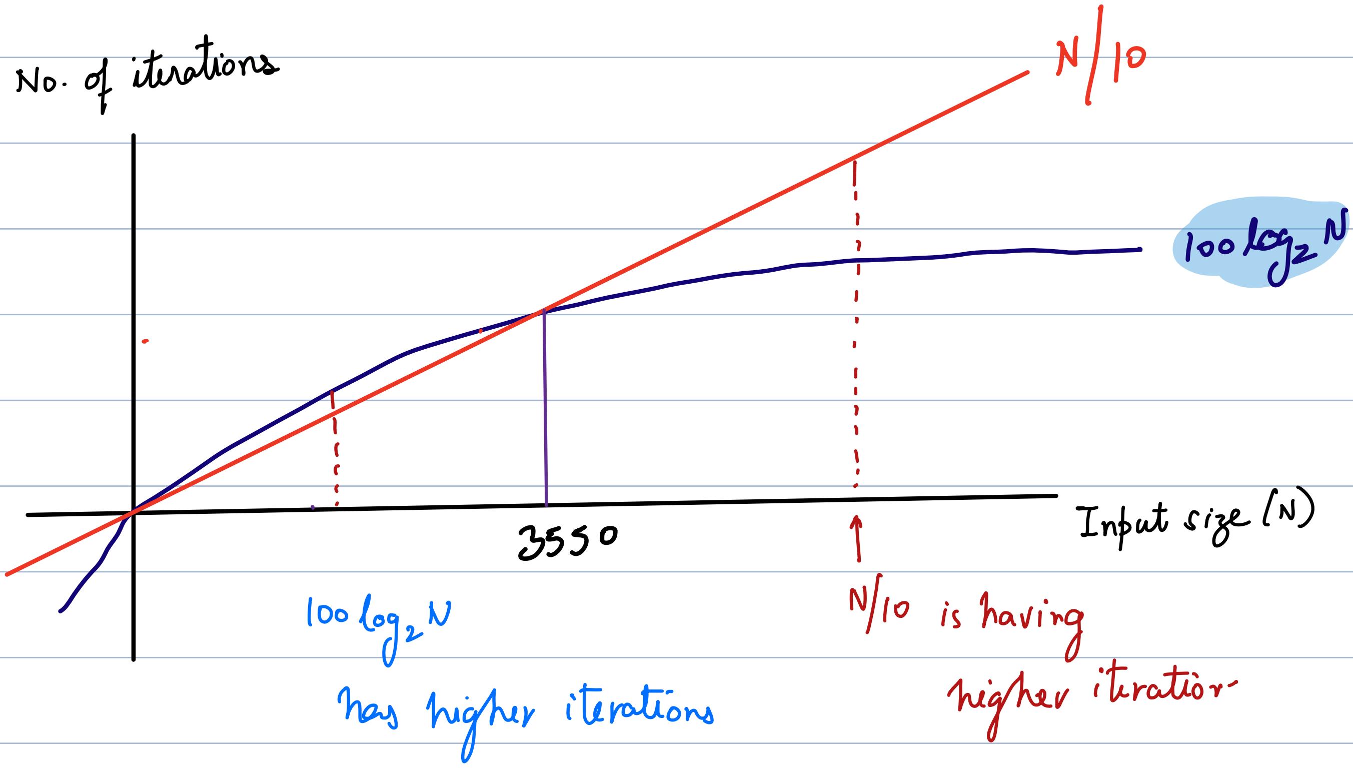
```
for (i=1; i<=N; i++){        [1, N]
    print(i)
}
```

2) No. of iterations is a better metric to analyse algorithms

Aadarsh	Karan
$100 \log_2 N$	$\frac{N}{10}$

$N = 2$

$100 \log_2 2$	$\frac{2}{10}$
100	0



below 3550 $N < 3550$

$\frac{N}{10}$ is performing better

above 3550 $N > 3550$

$100 \log_2 N$ performs better

N → Input size

2.2 Crore

Analyse algos for big values of N

Asymtotic analysis of algos

Analysing algos for big values of N

$N \to \infty$

Big O

1) Count no. of iterations
2) Take the highest order term
3) Remove constant coefficient

$N^3 + 3N^2$

$O(N^3)$

$5N + 1000$

$O(N)$

✓

$6N^2 + 3N^3 + N \times C$

$\downarrow$

$O(N^3)$

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

✓

$O(N^2)$ $\qquad\qquad$ $O(N^3)$

$N^2 + \boxed{N}$

$N = 10 \rightarrow 110$ $\qquad$ $\frac{10}{110} \times 100 = 9\%$

$N = 1000 \rightarrow 1001000$ $\qquad$ $\frac{1\cancel{000}}{1001\cancel{000}} \times 100$

$\frac{100}{1001} = 0.09\%$

We ignore lower order terms because their contribution is negligible

$N + 10^{10}$ $\qquad\qquad$ N^2

✓

$N = 10^{10}$

2×10^{10} $\qquad\qquad$ 10^{20}

✓

100 1000

$O(1)$ $O(1)$

Issues in big O

N^2 $3N^2$

better

$O(N^2)$ $O(N^2)$

1) You cannot compare algos with same big O value

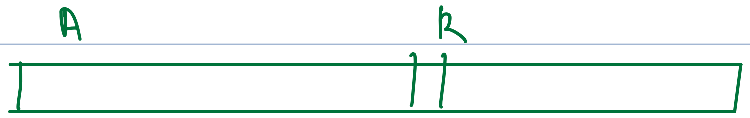
big O analysis is TC

.

N^2	$10N$
	✓
$N = 3$	
9	30

Issue

2) You cannot always say that O will work for all input values of N
(Might fail for smaller values of N)



{1, 2, 3, 4}

k=1

```
bool search (A, k) {
    for (i=0; i< len(A); i++) {
        if (A[i] == k) {
            return true
        }
    }
    return false
}
```

N is len of array

TC: O(N)

worst case

10:02 – 10:15

Space Complexity

Extra space required for your algorithm

```
func (int N){
    int x;    // 4
    int y;    // 4
    long z;   // 8
}
```

int : 4bytes

long : 8 bytes

16 bytes

SC: O(1)

```
func (int N){
    int x = N              : 4
    int y = x*x            : 4
    long z = x+y           : 8
    int arr = new int [N]  : 4xN
}
```

$4N + 16$

SC: O(N)

```
func (int N){
    int x = N                  4
    int y = x*x                4
    long z = x+y               8
    int arr = new int [N]      4N
    int l = new long [N][N]    8xN^2
}
```

3×3

long

$16 + 4N + 8N^2$

SC: $O(N^2)$

–	–	–
–	–	–
–	–	–

$8 \times N^2$

```
bool search (A, k) {
    for (i=0; i< len(A); i++) {
        if (A[i] == k) {
            return true
        }
    }
    return false
}
```

SC: O(1)

```
[] random ( N) {
    int A[N]
    for (i=0; i<N; i++) {
        A[i] = i
    }

    return A
}
```

TC: O(N)

SC: O(1)

Never count input and output of an algo while calculating SC prefix sum array

SC: O(1)

```
int [] prefix (A, N) {
    pref [N]
    for (i = 1 ; i < N ; i++) {
        pref [i] = pref [i-1] + A [i]
    }

    return pref
}
```

SC: O(N)

```
int [] prefix (A, N) {
    pref [N] , pref2 [N]
    for (i = 1 ; i < N ; i++) {
        pref [i] = pref [i-1] + A [i]
        pref2 [i] = pref [i]
    }
    return pref
}
```

SC: O(N)

```
void prefix (A, N) {
    pref [N]
    for (i = 1 ; i < N ; i++) {
        pref [i] = pref [i-1] + A [i]
    }
}
```

Time limit exceeded

Aakash —Referral→ Google

Pinklesh
↓
Google

1) Coding contest

Q2)

Q1) ___ → TLE ☹

↓ optimize

☹

$\boxed{1 \text{ second} \rightarrow 10^8 \text{ iteration}}$ Budget

1GHz processor → 10^9 operations in 1 second

Constraints

$1 \le N \le 10^3$

TC: $O(N^2)$

iterations: $N = 10^3 \Rightarrow 10^6$ ✓

TC : $O(N^3)$

iterations : $N = 10^3 \Rightarrow 10^9$ ∠

Done!

$2^{10} \approx 1024 - 1000$

$2^{30} = (1000)^3 = 10^9$

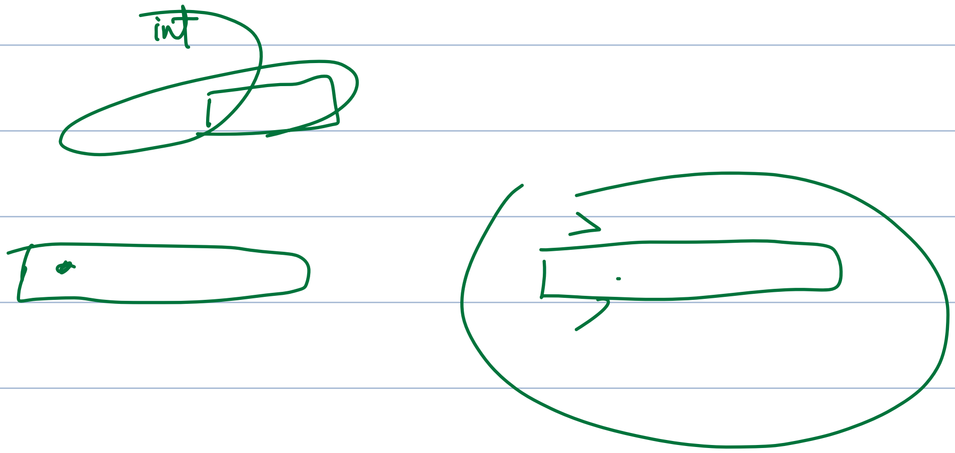
$i=1 \quad ; \quad i * i * i \le N$

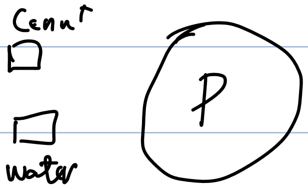
$i^3 \le N$

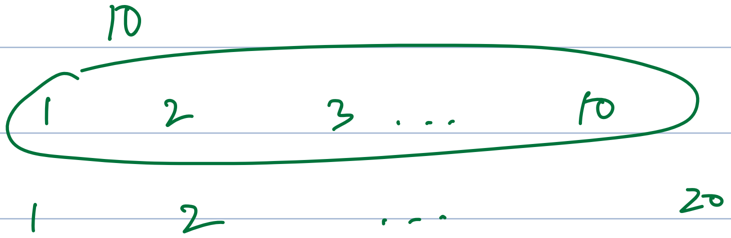
$i \le \sqrt[3]{N}$

$[1, \sqrt[3]{N}]$

itr. $O(N^{1/3})$




Cement
water
P


10
1 2 3 . . . 10
1 2 . . . 20