Time complexity

Sunday, 21-04-2021 7:02 AM

Check Point : 7 years

Sept (N) : 7 days

15 days

1 week

Pre. Year

Pre. Month

(Feb)

5s

4 + 2s (C++)

3s

4 + 3s

7s

5s

8s

5s

Life of operator

```
for (i=0; i<n; i++) {
    for (j=0; j<n; j++) {
        Print (n*i+j);
    }
}
```

$2n^2 + 1(n) + n$

$\Rightarrow f(n)$

$\Rightarrow O(n)$

Arithmetic Operator	+ , * , - , / , %
Comparison Operator	== , <= , >= , < , > , !=
Assignment Op	a=3 , b='c' ; a=a+1
Bitwise operator	&, |, ^, ~, <<, >>
Logical operator	&& , ||
Unary Operator	a++ , ++a , a--, --a
Function call	f (x)
Array operator	a[i] , a[i][j]
Pointer	*p , p->, &a

Simple Operator

```
for (i=0; i<n; i++) {
    for (j=0; j<n; j++) {
        Print (a[i][j]);
    }
}
```

$i + n(n) + n + n*(n+1)$

$f(n) = \boxed{4n^2 + 4n + 2} \Rightarrow O(n)$

$\boxed{f(n) = 4n^2 + \sqrt{n} + 3n + (\log n)^2} \Rightarrow 4n^2 \Rightarrow n^2$

$f_1(n) = 2n^2 + n \log n + 8n \Rightarrow 2n^2 \Rightarrow n^2$

2 km — 2000 m

5 km — 1 km

1 km — 50 m

200 m — 1.5 m

10 km

USA

12 km

12,000 km

10 km

→ Not execution time

→ It is the rate of increase of time taken w.r.t the input size.

Walking | Drive | Train

Mother f

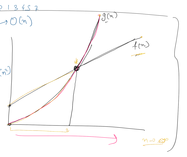
Father f

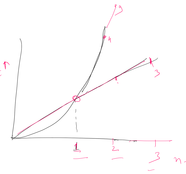
703 f2

2013 f53

$f(n) = 4n + 1 \Rightarrow n \rightarrow O(n)$

$g(n) = 2n^2 \Rightarrow n^2 \rightarrow O(n)$





Time Complexity

→ The measure of rate of growth of time w.r.t the input size.

Asymptotic Notations

Big - O

$O(n)$

① `for (i=0; i<n; i++) { if (i % 2 == 1) continue; }` → $O(n)$

② `for (i=0; i<n; i+=2) { Print (a[i]); }` → $f(n) = \frac{n}{2} + 1$ → $O(n)$

Simple (Best) — $O(n)$

$2x$ → $2x^2$ → $2n$

③ `for (i=1; i<n; i*=2) { Print (a[i]); }` → 13 → k → 1

$i = 1, 2, 4, 8, 16, \ldots, n$

$2^k = n$

$k = \log n$

$\log n$

$i = 1, 2, 4, 8, 16, \ldots, n$

$2^0, 2^1, 2^2, 2^3, 2^4, \ldots, 2^k$

$2^k = n$

$k = \log n$

$\boxed{\log(2^k) = k \log 2}$

$\boxed{\log n = k \log 2}$

$\boxed{k = \frac{\log n}{\log 2}}$

$O(\log n)$

$i = 1, 2, 4, 8, 16, \ldots, n$

$2^0, 2^1, 2^2, 2^3, 2^4, \ldots, 2^k$

$2^k = n$

$\boxed{2^k = n}$

$\boxed{\log 2^k = \log n}$

$\boxed{k \log 2 = \log n}$

$\boxed{k = \frac{\log n}{\log 2}}$

$\log_2 n$

$\log_2 n$

Q

Simple

```
int k = 0;
for (i=0; k<x; i++) {
    k = k + i;
}
```

(x > 0) — x start → Sept (N)

k	i
0	0
1	1
1+1	2
1+1+3	3
1+1+3+4	4
⋮	⋮
1+2+3 ... x	x

x start

K < N

$1 + 2 + 3 + \ldots + x = N$

$\frac{x(x+1)}{2} = N$

$\boxed{\frac{x^2 + x}{2} = N} \Rightarrow x^2 + x - 2N = 0$

$x = \sqrt{N}$

$\Rightarrow x^2 \approx N$

$x \approx \sqrt{N}$

Q

```
for (i=0; i<n; i++) {
    Print (Al[i]);
}
```

$\Rightarrow cn$

```
for (i=0; i<n; i++) {
    for (j=i; j<n; j++) {
        Print (a[i], a[j]);
    }
}
```

$\rightarrow C_1 n + C_2 \left(\frac{n^2 + n}{2}\right)$

$O(n^2)$

i = 0	n
i = 1	n-1
i = 2	n-2
i = 3	n-3
⋮	
i = n-1	1
i = n	0

$\Rightarrow 0 + 1 + 2 \ldots (n-2) + (n-1) + n$

$\frac{n \times (n+1)}{2}$

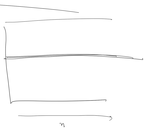
$= \frac{n^2 + n}{2}$

```
add (a, b)
    int c = a + b;
    return c;
```

$\Rightarrow 2c$

```
int Find (Arr[]) {
    int Arr[0];
}
```

$O(1)$



$O(1)$ | $O(\log n)$ | $O(\text{sqrt}(n))$ | $O(n)$ | $O(n \log n)$ | $O(n^2)$ | ... | $O(n^3)$ | ... | $O(2^n)$ | $O(n!)$ | $O(n^n)$

→ Concept → Problem

→ Problems → Bookwork

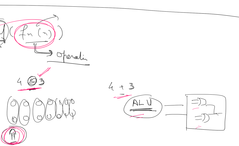
→ Workbook

1 2 3 4 — 10

$2^n \Rightarrow 2 \times 2 \times 2 \times 2 \ldots 2$

$n! \Rightarrow 2 \times 3 \times 4 \times 5 \ldots n$

$n^2 \Rightarrow n \times n \times n \times n \ldots n$

1 — fun (x) → Operands

4 + 3