L4- Asymptotic Notation

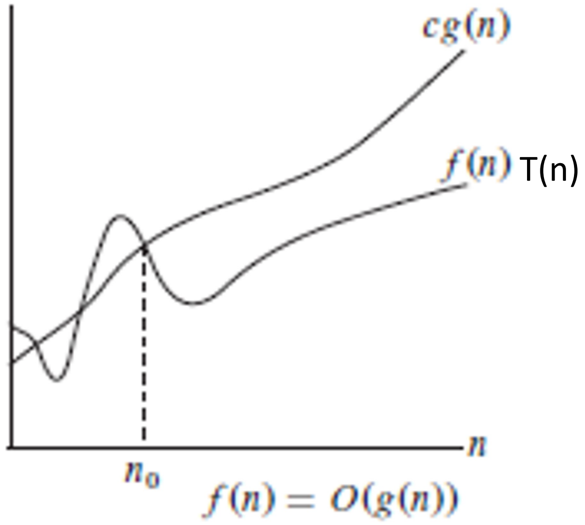
Monday, August 17, 2020 9:46 AM

O-notation

For a given function g(n), we denote by ***O(g(n))*** (pronounced as "big-oh of g of n") the set of functions

O(g(n)) = { T(n): there exists two positive constants c and n_0 such that
0 <= T(n) <= cg(n) for all n >= n_0 }

i.e. for all n greater than or equal to *n_0* , the graph of function ***T(n)*** lies on or below that of ***cg(n)***. In other words, *O(g(n))* is used to describe the ***Asymptotic Upper Bound*** on *T(n)*.



T(n) = 2*n +2*n + 1+1
= 4*n +2

We have to find g(n) such that 0<= T(n) <= c(g(n)) for all n >= n_0 for a particular c.

Let's assume g(n) = n
4n+2 <= n
Let's assume c=5
4n+2 <= 5n
If n=0,
2<=0

If n=1,   6 <= 5

If n=2,   10 <= 10 $n_0 = 2$
If n=3,   14 <= 15
If n=2000,   8002 <= 10000

Let's assume c=6
   4n+2 <= 6n

If n=0,   2<=0

If n=1,   6 <= 6 $n_0 = 1$
If n=2,   10 <= 12
If n=1000,   4002 <= 6000

T(n) = O(n)   O(n) = {T(n)}

$T_2(n)$ = 2*(n-1) +2*(n-1) +1+1
        = 4*n - 2
0<= $T_2(n)$ <= c(g(n))

Let g(n) = n
4n-2 <= n

If n=1,   2<=1
If n=2,   6<=2

Let's assume c=2
4n-2 <=2n

If n=1,   2<=2
If n=2,   6<=4

Let's assume c=4
4n-2 <=4n

If n=1,   2<=1 n_0=2
If n=2,   6<=8
If n=2000,   7998<=8000

Asymptotic Complexity of $T_2(n) = O(n)$

```
                                                  Operations(Cost)  *  Freq
//for (i=1 to n step 1)
for (int i=1; i<=n; ++i) {                              c1              n+1
   for (int k= 1; k <= n; ++k){                         c2              n*(n+1)
         a = a * k;                                     c3              n*n
   }
}
```

$T(n) = c1*(n+1) + c2*(n(n+1)) + c3(n*n)$

$\quad\;\; = n^2*(c2+c3) + n*(c2+c1) + c1$

$T(n) = n^2*(c10) + n*(c11) + c1$

Let $g(n) = n^2$

$n^2*(c10) + n*(c11) + c1 <= n^2$

$T(n) = O(n^2)$

If n=10, $100(c10) + 10(c11) + c1$

If n=100, $10000(c10) + 100(c11) + c1$

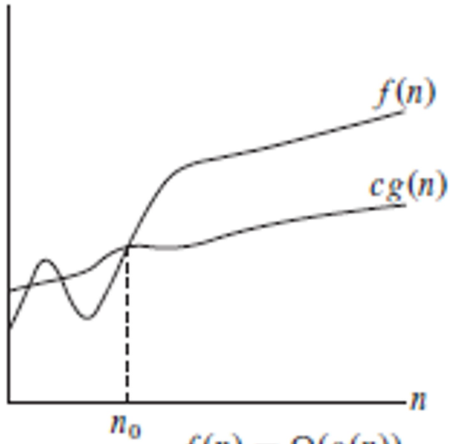
If n=1000, $1000000(c10) + 1000(c11) + c1$

Ω-notation

For any function g(n), Ω(g(n)) (pronounced as "big-Omega of g of n") denotes the set of functions

Ω(g(n)) = {T(n): there exists two positive constants c and n_0 such that
$$\mathbf{0} \le \boldsymbol{cg(n)} \le \boldsymbol{T(n)} \ \textbf{for all } \boldsymbol{n \ge n_0} \ \mathbf{\}}$$

$\Omega(g(n))$ *provides* ***asymptotic lower bound*** *on* $T(n)$ *i.e. the function* $cg(n)$ *is less than or equal to or the graph of* $cg(n)$ *lies at or below that of* $T(n)$ *for all* $n \ge n_0$.



$f(n) = \Omega(g(n))$

Selection Sort:
Best Case:

$$T(n) \ = \ 5(n-1) + \frac{2(n-1)n}{2}$$
$$= \ n^2 + 4n \ - 5$$

$$cn^2 \leq \ n^2 + 4n \ - 5$$
$$c \leq 1 + \frac{4}{n} - 5/n^2$$

If n=2
c <= 1+2-5/4
If n=10
c <= 1+.4-5/100
If n=100
c<= 1+.04-.0005

T(n) = $\Omega(n^2)$ for c=1 and $n \geq 2$

Θ-notation

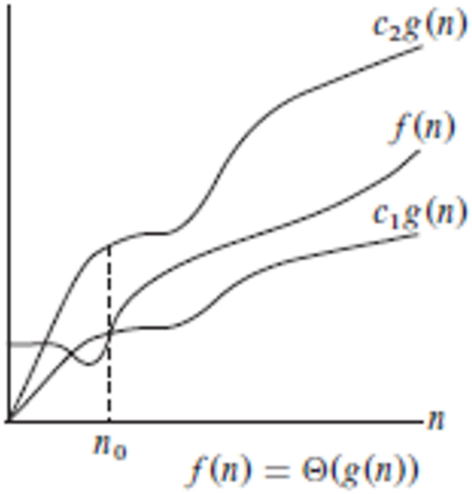
For any function g(n), Θ(g(n)) (pronounced as "theta of g of n") denotes the set of functions

Θ(g(n)) = {T(n): there exists three positive constants $\boldsymbol{c_1, c_2}$ and $\mathbf{n_0}$ such that

$$\mathbf{0} \leq \boldsymbol{c_1 g(n) \leq T(n) \leq c_2 g(n)} \text{ for all } \boldsymbol{n \geq n_0} \}$$

$\Theta(g(n))$ *provides* ***asymptotic tight bound*** *on* $T(n)$ *i.e. the function* $T(n)$ *is* candwiched between $c_1 g(n)$ and $c_2 g(n)$ for all $\mathrm{n} \geq \mathrm{n}_0$.

Th: When $\mathrm{T(n)} = \mathrm{O(g(n))}$ and $\mathrm{T(n)} = \Omega(g(n))$, then $\mathrm{T(n)} = \Theta(g(n))$



For Selection Sort, the Time Complexity is $\Theta(n^2)$. Same is the case for Bubble Sort.