Kinds of Analysis of Algorithms

• **The Priori Analysis** is aimed at analyzing the algorithm before it is implemented(based on the algorithm) on any computer. It will give the approximate amount of resources required to solve the problem before execution. In case of priori analysis, we ignore the machine and platform dependent factors. It is always better if we analyze the algorithm at the earlier stage of the software life cycle.
Priori analysis require the knowledge of

– Mathematical equations
– Determination of the problem size
– Order of magnitude of any algorithm

• **Posteriori Analysis** is aimed at determination of actual statistics about algorithm's consumption of time and space requirements (primary memory) in the computer when it is being executed as a program in a machine.
Limitations of Posteriori analysis are
• External factors influencing the execution of the algorithm
– Network delay
– Hardware failure etc.,
• The information on target machine is not known during design phase
• The same algorithms might behave differently on different systems
– Hence can't come to definite conclusions

Asymptotic notations(O,Ω,Θ)

Step count is to compare time complexity of two programs that compute same function and also to predict the growth in run time as instance characteristics changes. Determining exact step count is difficult and not necessary also. Since the values are not exact quantities we need only comparative statements like $c_1n^2 \leq t_p(n) \leq c_2n^2$.

For ex: consider two programs with complexities $c_1n^2 + c_2n$ and c_3n respectively. For small values of n, complexity depend upon values of c_1, c_2 and c_3. But there will also be an n beyond which complexity of c_3n is better than that of $c_1n^2 + c_2n$.This value of n is called break-even point. If this point is zero, c_3n is always faster (or at least as fast).

c1=1,c2=2 & c3=100

Then $c_1n^2+c_2n$ is $\leq c_3n$ for $n\leq 98$ and

$c_1n^2+c_2n$ is $> c_3n$ for $n>98$

The Common asymptotic functions are given below.

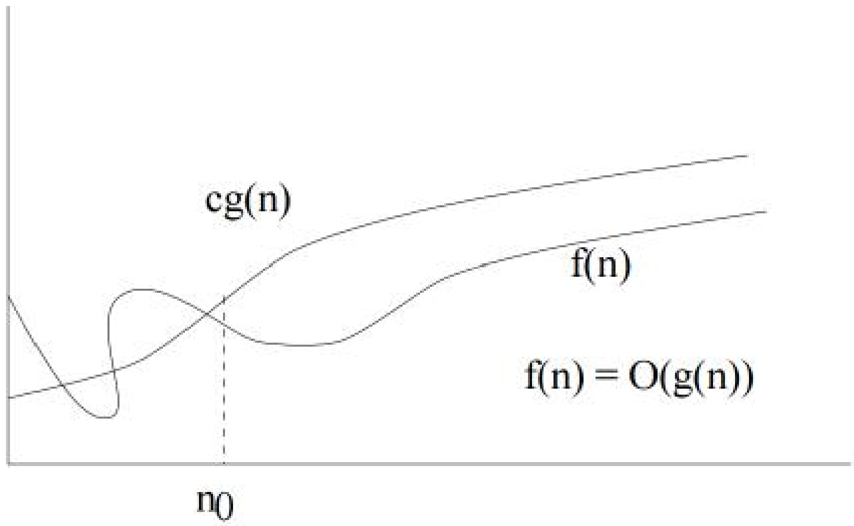
Function	Name
1	Constant
log n	Logarithmic
N	Linear
n log n	n log n
n^2	Quadratic
n^3	Cubic
2^n	Exponential
n!	Factorial

The growth of the functions as below

$$1< \log n < n <n \log n < n^2 < n^3 < 2^n <n!$$

Definition [Big 'oh']: If f(n) is a nonnegative, then function $f(n)=O(g(n))$ iff there exist positive constants c and no such that $f(n)\leq c*g(n)$ for all n, $n \geq$ no.

- We say that "f(n) is big-O of g(n)."
- As n increases, f(n) grows no faster than g(n).
- In other words, g(n) is an asymptotic upper bound on f(n).



Ex1: $f(n) = 2n + 8$, and $g(n) = n^2$. Can we find a constant c, so that $2n + 8 <= n^2$? The number 4 works here, giving us 16 <= 16.
For any number c greater than 4, this will still work. Since we're trying to generalize this for large values of n, and small values (1, 2, 3) aren't that important, we can say that f(n) is generally faster than g(n); that is, f(n) is bound by g(n), and will always be less than it.

Ex2: The function $3n+2=O(n)$ as $3n+2\leq4n$ for all $n\geq2$.

Pb1: 3n+3=O(_______) as 3n+3≤______ for all________.

Ex3: $10n^2+4n+2=O(n^2)$ as $10n^2+4n+2\leq11n^2$ for all $n\geq5$

Pb2:$1000n^2+100n-6$=O(_______) as $1000n^2+100n-6\leq$________for all_______

Ex4:$6*2^n+n^2=O(2^n)$ as $6*2^n+n^2 \leq 7*2^n$ for $n \geq 4$

Ex5:$3n+3=O(n^2)$ as $3n+3\leq3n^2$ for $n\geq2$.

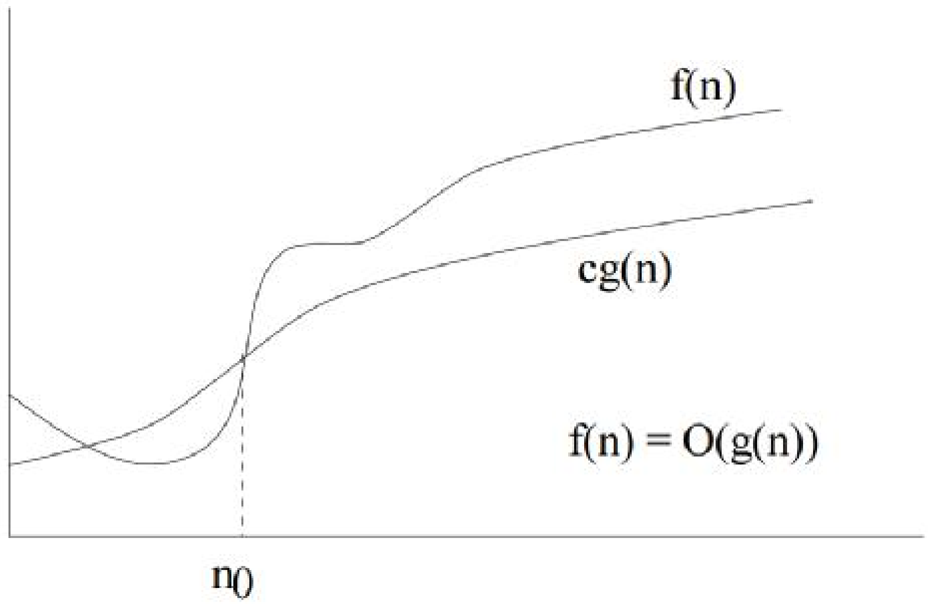
Ex 6:$10n^2+4n+2$=On4 as$10n^2+4n+2\leq10n^4$ for $n\geq2$.

Ex7:$3n+2\neq O(1)$ as3n+2 not less than or equal to c for any constant c and all $n\geq n_0$

Ex 8: 10n2+4n+2≠O(n)

Definition[Omega] The function $f(n) = \Omega(g(n)$ (read as "f of n is omega of g of n") iff there exist positive constants c and n_0 such that $f(n) \geq c * g(n)$ for all n, $n \geq n_0$.

- We say that "f(n) is omega of g(n)."
- As n increases, f(n) grows no slower than g(n).
- In other words, g(n) is an asymptotic lower bound on f(n).



Ex1: The function $3n+2=\Omega(n)$ as $3n+2 \geq 3n$ for $n \geq 1$
(the inequality holds for $n \geq 0$,but the definition of Ω requires an $n0>0$).
$3n+2=\Omega(n)$ as $3n+2 \geq 3n$ for $n \geq 1$.

Ex:2 $100n+6=\Omega(n)$ as $100n+6 \geq 100n$ for $n \geq 1$

Pb:1 $6n+4= \Omega($_____$)$ as $6n+4 \geq$_______ for all $n \geq$ _____

Ex:3 $10n^2 +4n+2= \Omega(n^2)$ as $10n^2 +4n+2 \geq n^2$ for $n \geq 1$

Pb:2 $2n^2+3n+1= \Omega($ _____ $)$ as $2n^2+3n+1 \geq$_______ for all $n \geq$_________

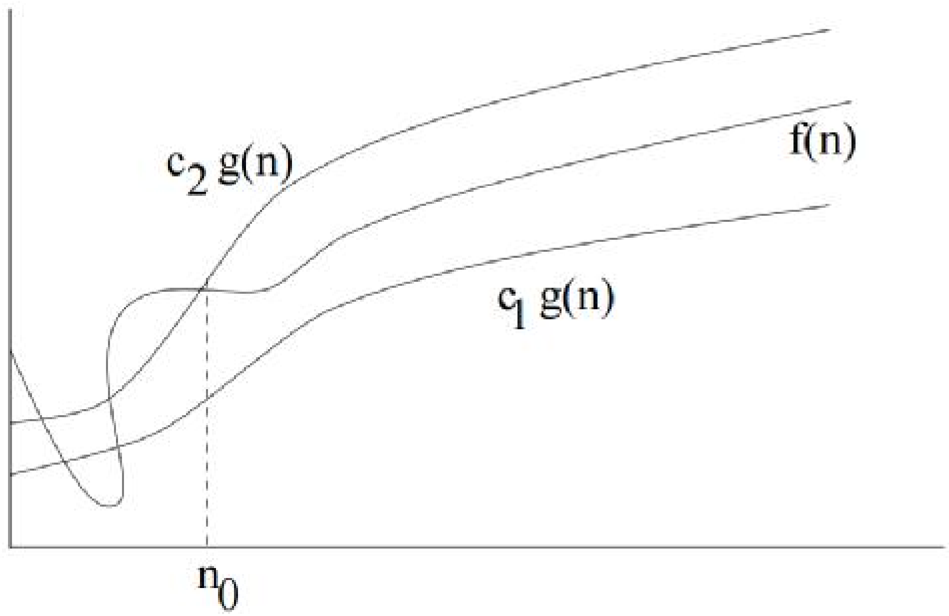
Ex: 4 $6* 2^n + n^2 = \Omega(2^n)$ as $6* 2^n + n^2 \geq 2^n$ for $n \geq 1$.

Pb:3 $4n^2 - 64 n + 288 = \Omega$ (____)as $4n^2 - 64 n + 288 \geq$_______ for all $n \geq$_________ .

Pb:5 $n^3 \log n$ is $= \Omega$ (____)as $n^3 \log n$__________for all $n \geq$ ___.

Definition [Theta] The function $f(n) = \Theta(g(n))$ (read as "f of n is theta of g of n") iff there exist positive constants C_1, C_2, and n_0 such that $c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all n, $n \geq n_0$

- We say that "f(n) is theta of g(n)."
- As n increases, f(n) grows at the same rate as g(n).
- In other words, g(n) is an asymptotically tight bound on f(n).



Ex 1 : The function $3n + 2 = \Theta(n)$ as $3n + 2 \geq 3n$ for all $n \geq 2$ and $3n + 2 \leq 4n$ for all $n \geq 2$ so $c_1 = 3$, $c_2=4$ and $n_0 = 2$

Pb 1: $6n+4= \Theta($___$)$ as $6n+4 \geq$________for all $n \geq$_______and $6n+4 \leq$_______ for all $n \geq$_______

Ex 2 : $3n + 3 = \Theta(n)$

Ex 3 : $10n^2 + 4n + 2 = \Theta(n^2)$

Ex 4: $6 * 2^n + n^2 = \Theta(2^n)$

Ex5: $10 * \log n + 4 = \Theta(\log n)$

Ex6: $3n+2 \neq \Theta(1)$

Ex7: $3n + 3 = \Theta(n)$

Ex 8: $10n^2 + 4n + 2 \neq \Theta(n)$

Theorem: If $f(n) = a_m n^m + \ldots\ldots\ldots\ldots\ldots\ldots + a_3 n^3 + a_2 n^2 + a_1 n + a_0$, then $f(n) = O(n^m)$

Proof: $f(n) \le \sum_{i=0}^{m} |a_i| n^i$

$\le n^m \sum_{i=0}^{m} |a_i| n^{i-m}$

$\le n^m \sum_{i=0}^{m} |a_i|$ for $n \ge 1$

$f(n) = O(n^m)$ (assuming that m is fixed).

Theorem: If $f(n) = a_m n^m ++ \ldots\ldots\ldots\ldots + a_3 n^3 + a_2 n^2 + a_1 n + a_o$ and $a_m > 0$, then $f(n) = \Omega(n^m)$

Proof : Left as an exercise.

Definition [Little 'oh'] The Function f (n) =o(g(n)) (read as 'f of n is little oh of g of n')iff

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

Example:

The function $3n+2 = o(n^2)$ since $\lim_{n\to\infty} \frac{3n+2}{n^2} = 0.$

Ex1: $3n+2 = o(n \log n)$.

Ex:2 $3n+2 = o(n \log \log n)$.

Ex:3 $6*2^n + n^2 = o(3^n)$.

Ex:4 $6 * 2^n + n^2 = o(2^n \log n)$.

Ex:5 $6 * 2^n + n^2 \neq o(2^n)$.

Analogous to 'o' notation ' ω 'notation is defined as follows.

Definition [Little omega] The function f(n)= ω(g(n)) (read as " f of n is little omega of g of n") iff

$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$$