Key

n = the original input size
T(n) = time required to process input of size n
F = factor by which the input size is multiplied
T(Fn) = time required to process input of size Fn

O(log n) Algorithms

- It doesn't matter which base you use for the logs, as long as you use the same base throughout the problem.
- If the numbers in the problem are powers of 10, use base 10.
- If the numbers in the problem are powers of 2, use base 2.
- Even if you don't use the base suggested above, you'll still get the same answer, but the arithmetic will be more complicated.

T(n) = c log(n)
T(Fn) = c log(Fn)

$$\frac{T(Fn)}{T(n)} = \left(1 + \frac{\log(F)}{\log(n)}\right)$$

$$\frac{T(n)}{\log(n)} = \frac{T(Fn)}{\log(Fn)}$$

O(n) Algorithms

T(n) = cn
T(Fn) = c(Fn)

$$\frac{T(Fn)}{T(n)} = F$$

$$\frac{T(n)}{n} = \frac{T(Fn)}{Fn}$$

O(n log n) Algorithms

T(n) = c n log(n)
T(Fn) = c (Fn) log(Fn)

$$\frac{T(Fn)}{T(n)} = F\left(1 + \frac{\log(F)}{\log(n)}\right)$$

$$\frac{T(n)}{n\log(n)} = \frac{T(Fn)}{(Fn)\log(Fn)}$$

O(n^2) Algorithms

T(n) = c n^2
T(Fn) = c $(Fn)^2$

$$\frac{T(Fn)}{T(n)} = F^2$$

$$\frac{T(n)}{n^2} = \frac{T(Fn)}{(Fn)^2}$$

O(n^3) Algorithms

T(n) = c n^3
T(Fn) = c $(Fn)^3$

$$\frac{T(Fn)}{T(n)} = F^3$$

$$\frac{T(n)}{n^3} = \frac{T(Fn)}{(Fn)^3}$$

O(2^n) Algorithms

T(n) = c 2^n
T(Fn) = c $2^{(Fn)}$

$$\frac{T(Fn)}{T(n)} = 2^{(F-1)n}$$

$$\frac{T(n)}{2^n} = \frac{T(Fn)}{2^{(Fn)}}$$