Thu 02/18/16

Master theorem

This theorem just tells us the the solution to the recurrences of form

T(1) = O(1)

T(n) = aT(n/b) + O(f(n))

where f(n) = $O(n^2)$

Ex. T(n) = 2T(n/2) + O(n)

a = 2, b = 2, c = 1

$\log_b a = \log_b 2 = 1$

T(n) = {

$O(n^{\log_b a})$ if $c < \log_b a$

$O(n^{\log_b a})$ if $c = \log_b a$

$O(n^c)$ if $c > \log_b a$

}

```
Ex T(n) = 2T(n/2) + O(sqrt(n) logn)
a = 2, b = 2, log(b)(a) = 1
c = 1/2 + epsilon eg. c = 2/3
sqrt(n) logn = O(n^(2/3)) => T(n) = O(n)
```

Proof: Assume f(n) <= αn^2 for all n>=1

T(n) <= $\alpha T(n/b) + \alpha n^c$

T(n/b) <= $at(n/b^2) + \alpha(n/b)^2$

T(n) <= $a[aT(n/b^2)] + \alpha(n/b)^c] + \alpha n^c$

$T(n/b^2)$ <= $aT(n/b^3) + \alpha(n/b^2)^c$

T(n) <= $a^2[aT(n/b^3) + \alpha(n/b^2)^c] + \alpha n^c(1+a/b^2)$

= $a^3T(n/b^2) + \alpha(a^2/b^{2c})n^c + \alpha n^c(1 + a/b^c)$

= a^3T(n/b^3) + αn^c[1 + a/b^c + (a/b^2)^2)]

= a^i T(n/b^i) + αn^c[Sum of all from j=0 to i-1 (a/b^c)^j] n/b^i <= 1

$n <= b^i$ $i >= log_b$

Logarithmic Timeout

1. $a^{log_a n} = n$
2. $log_b n^2 = clog_b n$
3. $log_b n = log_a n / log_a b$
4. $n^{log_b a} = n^{log_b}$