Dhantha Gunarathna

Home work 1

January 12 2016

1.10

For T(n) to be O(f(n)), f(n) should be the upper bound of the growth rate of f(n) should be the upper bound of the growth rate of T(n)

T(n) is Omega(g(n)) mean there exists T(n) >= c*g(n)

When i=1

f1(n) = n^2 is O(n^2)

f2(n) = n^2 + 1000*n is O(n^2)

f3(n) is Omega(f3(n))

f(4) is O(n^3)

1.12 Using Big Oh notation, the worst case running times

a. n^3*t is O(n^3)

b. t*j*n-(i+1)*(n-1) is O(n^2)

c. j*(n-i)*n is O(n^2)

d. T(n) = 2*T(n-1)

T(1) = 2*T(0)

T(2) = 2*T(1) = 4*T(0)

T(3) = 2*T(2) = 8*T(0)

Therefore T(n) = 2*T(n-1) is O(2^n)