1. 8n2 <= 64nlgn

n2 <= 8nlgn

n <= 8lgn

Insertion sort will be faster for values of n <≈ 43

1. 1. f(n) = O(g(n)); For any given n where n > 0, n0.25 will always be less than n0.5
   2. f(n) = Ω(g(n)); log2n = (logn)2
   3. f(n) = Ω(g(n)); log n = (ln n)/(ln 10) = ln n / 2.303
   4. f(n) = O(g(n)); limn->∞ f(n)/g(n) = 1000n2 / (0.0002n2 – 1000n)

= n(1000n) / n(0.0002n – 1000) = 1000n / (0.0002n – 1000)

= 5000000n / n – 500000

* 1. f(n) = Ω(g(n)); n√n = n3/2 = √n3 “>” nlogn
  2. f(n) = O(g(n)); e ≈ 2.718 < 3
  3. f(n) = Θ(g(n)); limn->∞ f(n)/g(n) = 2n / 2n+1 = 2n / (2(2n)) = ½
  4. f(n) = Ω(g(n)); limn->∞ f(n)/g(n) = 2n / 22n = 2n / (22)n = 2n / 4n = limn->∞ (1/2)n = ∞
  5. f(n) = O(g(n));
  6. f(n) = O(g(n)); lg n = 2lg √n

1. 1. Suppose f1(n) = n2 and g(n) = n3. Thus, f1(n) = O(g(n)) => n2 = O(n3). Also, suppose f2(n) = n. Thus, f2(n) = O(g(n)) => n = O(n3). Therefore, f1(n) = O(f2(n)) => n2 = O(n). Thus, we have a contradiction.
   2. Suppose f1(n) = n and g1(n) = n2. Thus, f1(n) = O(g1(n)). Suppose f2(n) = lg n and   
      g2(n) = n lg n. Thus, f2(n) = O(g2(n)). Therefore, by substitution, we have  
      n + lg n = Θ(n2 + n lg n), which is incorrect. Thus we have a contradiction, proving the original statement to be incorrect.