Todd Wenker

CSE 310

Professor Nakamura

1/28/15

1. For each of the following pairs of functions f(n) and g(n), determine only one of f(n) = O(g(n)), f(n) = Ω(g(n)), or f(n) =θ(g(n)). If f(n) =θ(g(n)), then do not choose f(n) = O(g(n)) or f(n) = Ω(g(n)).
   1. f(n) = 3n^2 + 5n^2(n^.5) + 2n(n^.5), g(n) = 6n^2(n^.5) + 8n(n^.5). f(n) = θ(g(n))
   2. f(n) = 5n^6, g(n) = (n^4 – 13n^3)/3. f(n) = Ω(g(n)).
   3. f(n) = log(n^3), g(n) = log(n^4). (logs are base 5). f(n) = θ(g(n)).
   4. f(n) = 8^n, g(n) = 5^n. f(n) = Ω(g(n)).
   5. f(n) = 4log n, g(n) = 3log n. (the first log is base 3, the second is base 4). f(n) = Ω(g(n)).
2. Suppose that the running time of the algorithm A is 990n^2, and the running time of the algorithm B is 60n^4. What is the largest value of n (a positive integer) for which the running time of the algorithm A is larger than that of B?

990n^2 = 60n^4

16.5^.5 = n^2

n = (16.5^.5)

When n is less than (16.5^.5) which is approximately 4.06, algorithm A is larger than B.

1. Prove that 9n^2 + 3n + 8 = O(n^2), using the definition of O notation.

(definition of O notation) 9n^2 + 3n + 8 <= an^2 for b, a > 0 and n >= b.

If b = 1 then n >= 1 and n^2 >=1. If n >= 1 then n^2 >= n

To find a: 9n^2 + 3n + 8 <= 9n^2 +3n^2 + 8n^2 = 20n^2 since n^2 >= n whenever n >= 1, setting a to 20 will ensure that

9n^2 + 3n + 8 <= 20n^2 whenever n >= 1

Because to this, we can conclude that 9n^2 + 3n + 8 = O(n^2).

1. Prove that 9n^2 - 3n - 10 = Ω(n^2) using the definition of Ω notation.

(definition of Ω notation) 9n^2 – 3n - 10 >= an^2 where a, b > 0 and n >= b

to be smaller than f(n), a needs to be less than 9

if a = 2 then 9n^2 – 3n -10 >= 2n^2 🡺 7n^2 – 3n – 10 >= 0 which factors into…

(n + 1) (7n – 10) >= 0

n <= -1 or n >= 10/7. Because b must be greater than 1

b = 10/7

a = 2 and b =10/7

if n >= b = 10/7, n – 10/7 >= or (7n – 10) >= 0. When b >= 10/7, the equation (n + 1) is also positive. As such, 9n^2 – 3n - 10 >= 2n^2 when n >= 10/7.

By the definition of Ω notation, 9n^2 – 3n - 10 = Ω(n^2).

1. Prove that if f(n) = O(g(n)) and h(n) = (f(n))^2 then h(n) = O((g(n))^2) using the definition of O notation.

The definition of O notation is f(n) <= a\*g(n) for a, b > 0 and n >= b.

One of the rules of O notation is that f(n) \* g(n) = O(f(n))\*O(g(n)).

Because h(n) = f(n)\*f(n), the O notation of h(n) is the O notation of f(n) squared which, in this case, is g(n)^2. As such, h(n) = O((g(n))^2).

1. Running time of Function (A, B) //m < n

|  |  |  |  |
| --- | --- | --- | --- |
| Row # | Function(A, B) // m < n | Constant | Times |
| 1) | m = length[A] | C1 | 1 |
| 2) | n = length[B] | C2 | 1 |
| 3) | i = 0 | C3 | 1 |
| 4) | success = false | C4 | 1 |
| 5) | while((i <= (n-m) && (success == false)) | C5 | Best: Worst:  2 (n-m) + 2 |
| 6) | j = 0 | C6 | Best: Worst:  1 (n-m) + 1 |
| 7) | success = false | C7 | Best: Worst:  1 (n-m) + 1 |
| 8) | while(( j < m) && (success == true)) | C8 | Best: Worst:  m+1 (m+1)(n-m + 1) |
| 9) | if(A[j] != B[i+j]) | C9 | Best: Worst:  m (m)(n-m + 1) |
| 10) | success = false | C10 | Best: Worst:  0 (n-m) + 1 |
| 11) | j = j + 1 | C11 | Best: Worst:  m (m)(n-m + 1) |
| 12) | i = i + 1 | C12 | Best: Worst:  1 (n-m) + 1 |
| 13) | return(i-1) | C13 | 1 |

Worst Case: Example

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

i=0 A B

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 1 |

A[5] !=B[5]: succ = false

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

i=1 A B

A[5] !=B[6]: succ = false

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

i=2 A B

A[5] !=B[7]: succ = false

I=3: stop

Best Case Example: A = the first m elements of B

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

i=0 A B

A[0-5] = B[0-5]: succ = true

i=1: stop

Worst Case:

T(m,n) = C1 + C2 + C3 + C4 + (n-m+2)C5 + (n-m+1)C6 + (n-m+1)C7 + (m+1)(n-m+1)C8 + (m)(n-m+1)C9 + (n-m+1)C10 + m(n-m+1)C11 + (n-m+1)C12 + C13

C8 is the faster growing term so:

(m+1)(n-m+1) = (mn-m^2+n+1) because n>m, (mn) is the fastest growing term so T(m,n) = θ(mn)

Best Case:

T(m,n) = C1 + C2 + C3 + C4 + 2C5 + C6 +C7 + (m+1)C8 + (m)C9 + mC11 + C12 + C13

C8 is the faster growing term: (m+1) = θ(m) so T(m,n) = θ(m)