Kobe Oley Due: 9/18/2020

CSCI 261: Homework 1

1. 22^(n+1)

22^(n)

(n+1)!

n!

en

n×2n

2n

(3/2)n

nlg(lg(n)) , (lg(n))lg(n)

(lg(n))!

n3

n2 , 4lg(n)

n×lg(n) , lg(n!)

n , 2lg(n)

(√2)lg(n)

2sqrt(2xlg(n))

lg2(n)

ln(n)

√(lg(n))

ln(ln(n))

2lg\*(n)

lg\*(lg(n)) , lg\*(n)

lg(lg\*(n))

1 , n1/lg(n)

1. ax

xc

k√(x)

logb(x)

1. 1. If n = f(n), then f(n) ∈ O(n2)

Consider N=1 & c=1

Suppose n ≥ 1

1 ≤ n => n ≤ n2

=> |n| ≤ |n2| since both results are positive numbers

=> |n| ≤ 1 \* |n2|

* 1. If n2 = f(n), then f(n) ∈ O(n2)

Consider N=0 & c=2

Suppose n ≥ 0

0 ≤ n => 0 ≤ n2

=> n2 ≤ n2 + n2

=> n2 ≤ 2 \* n2

=> |n2| ≤ 2 \* |n2| since both results are positive numbers

* 1. If 3n2 + 5n = f(n), then f(n) ∈ O(n2)

Consider N=1 & c=8

Suppose n ≥ 1

1 ≤ n => 5 ≤ 5n

=> 5n ≤ 5n2

=> 5n + 3n2 ≤ 5n2 + 3n2

=> 3n2 + 5n ≤ 8n2

=> |3n2 + 5n| ≤ |8n2| since both results are positive numbers

=> |3n2 + 5n| ≤ 8 \* |n2|

1. 1. Given that nΣk=2 (1/k) ≤ *ln*(n) – *ln*(1), prove that Hn ∈ O(*ln*(n))

Consider N = 1 & c = 2

Suppose n ≥ 1

1 ≤ n => n ≤ n2

=> *ln*(n) ≤ *ln*(n2)

=> *ln*(n/1) ≤ 2 \* *ln*(n)

=> *ln*(n) - *ln*(1) ≤ 2 \* *ln*(n)

=> nΣk=2 (1/k) ≤ *ln*(n) – *ln*(1) ≤ 2 \* *ln*(n)

=> nΣk=2 (1/k) ≤ 2 \* *ln*(n) due to the transitive law

=> |nΣk=2 (1/k)| ≤ |2 \* *ln*(n)| because both results are positive numbers

=> |nΣk=2 (1/k)| ≤ 2 \* |*ln*(n)|

* 1. Given that nΣk=2 (1/k) ≥ *ln*(n+1) – *ln*(2), prove that Hn ∈ Ω (*ln*(n))

Consider N = 1 & c = ½

Suppose n ≥ 1

n ≥ 1 => n - 1 ≥ 0

=> (n - 1)2 ≥ 0

=> n2 – 2n + 1 ≥ 0

=> n2 + 2n + 1 ≥ 4n

=> (n2 + 2n + 1)/4 ≥ (4n)/4

=> (n+ 1)2/4 ≥ n

=> ((n+ 1)/2)2 ≥ n

=> *ln*(((n+ 1)/2)2) ≥ *ln*(n)

=> 2 \* *ln*((n+ 1)/2) ≥ *ln*(n)

=> *ln*(n+ 1) – *ln*(2) ≥ *ln*(n)/2

=> nΣk=2 (1/k) ≥ *ln*(n+ 1) – *ln*(2) ≥ *ln*(n)/2

=> nΣk=2 (1/k) ≥ *ln*(n)/2 due to the transitive law

=> |nΣk=2 (1/k)| ≥ |*ln*(n)/2| due to both results being positive numbers

=> |nΣk=2 (1/k)| ≥ ½ \* |*ln*(n)|

1. The smallest n where I noticed that *fib* was running slow enough that it wasn’t negligible was when n = 28. That was when I noticed a significant difference in the speed of the function.
2. 1. Let n = 2

*f*(2, a, b) = *f*(1, b, a + b)

= a + b

= *f*(0, a, b) + *f*(1, a, b)

= *f*(n-2, a, b) + *f*(n-1, a, b)

Assume that the following is true: *f*(n, a, b) = *f*(n-1, a, b) + *f*(n-2, a, b)

Then when n = k + 1

*f*(k + 1, a, b) = *f*((k + 1) - 1, b, a + b)

= *f*(k, b, a + b)

= *f*(k - 1, b, a + b) + *f*(k - 2, b, a + b)

= *f*(k, a, b) + *f*(k - 1, a, b)

= *f*((k + 1) - 1, a, b) + *f(*(k + 1) - 2, a, b)

* 1. Let n = 0

*F*0 = 0

= *f*(0, 0, 1)

Let n = 1

*F*1 = 1

= *f*(1, 0, 1)

Assume that the following is true: *F*k = *f*(k, 0, 1) for 0 ≤ k < n

*F*n = *F*n-1 + *Fn-2*

= *f*(n-1, 0, 1) + *f*(n-2, 0, 1)

= *f*(n, 0, 1)

1. The *fibIt* function runs much faster than the fib function at higher values of n, such as at 28.