CSC6013 - Worksheet for Week 5

Back Substitution

Compute the complexity of the recursive algorithms based on the recursive equation and stop condition. Show your work, not just your final answer.

1. $T(n) = 2T(n-1) + 1$ and $T(0) = 1$
   a. You can compute this complexity as a tight upper bound.

<u>Substitute (n-1) –> k = 2</u>

$T(n-1) = 2T((n-1)-1) + 1$
$T(n-1) = 2T(n-2) + 1$

$T(n) = 2[2T(n-2) + 1] +1$
$\mathbf{T(n) = 4T(n-2) + 2 + 1}$

<u>Substitute (n-2) –> k = 3</u>

$T(n-2) = 2T((n-2)-1) + 1$
$T(n-2) = 2T(n-3) + 1$

$T(n) = 4[2T(n-3) + 1] + 2 +1$
$\mathbf{T(n) = 8T(n-3) + 4 + 2 + 1}$

<u>General Formula</u>

$\mathbf{T(n) = 2^k T(n-k) + 2^{k-1} + 2^{k-2} + \ldots + 2^1 + 2^0}$

<u>Stop Condition</u>

$T(0) = 1 \rightarrow n-k = 0 \rightarrow \mathbf{k = n}$

$T(n) = 2^n T(n-n) + 2^{n-1} + 2^{n-2} + \ldots + 2^1 + 2^0$
$T(n) = 2^n T(0) + 2^{n-1} + 2^{n-2} + \ldots + 2^1 + 2^0$
$\mathbf{T(n) = 2^n + 2^{n-1} + 2^{n-2} + \ldots + 2^1 + 2^0 = 2^{n+1} - 1}$

<u>Big-Oh Complexity</u>

$O(2^n)$

CSC6013 - Worksheet for Week 5

2. $T(n) = T(n-2) + n^2$ and $T(0) = 1$

a. Hint: Assume n is even; that is, n = 2k for some integer k.

<u>Substitute (n-2) –> k = 2</u>

$T(n-2) = T((n-2)-2) + (n-2)^2$
$T(n-2) = T(n-4) + (n-2)^2$

$\mathbf{T(n) = T(n-4) + (n-2)^2 + n^2}$

<u>Substitute (n-4) –> k = 3</u>

$T(n-4) = T((n-4)-2) + (n-4)^2$
$T(n-4) = T(n-6) + (n-4)^2$

$\mathbf{T(n) = T(n-6) + (n-4)^2 + (n-2)^2 + n^2}$

<u>Substitute (n-6) –> k = 4</u>

$T(n-6) = T((n-6)-2) + (n-6)^2$
$T(n-6) = T(n-8) + (n-6)^2$

$\mathbf{T(n) = T(n-8) + (n-6)^2 + (n-4)^2 + (n-2)^2 + n^2}$

<u>General Formula</u>

$\mathbf{T(n) = T(n-2k) + (n-2(k-1))^2 + (n-2(k-2))^2 + (n-2(k-3))^2 + \dots + (n-2)^2 + n^2}$

<u>Stop Condition</u>

$T(0) = 1 \rightarrow n-2k = 0 \rightarrow$ **k = n/2**

$T(n) = T(n-2(\frac{n}{2})) + (n-2((\frac{n}{2})-1))^2 + (n-2((\frac{n}{2})-2))^2 + (n-2((\frac{n}{2})-3))^2 + \dots + (n-2)^2 + n^2$
$T(n) = T(n-n) + (n-n+2)^2 + (n-n+4))^2 + (n-n+6))^2 + \dots + (n-2)^2 + n^2$
$T(n) = T(0) + (2)^2 + (4)^2 + (6)^2 + \dots + (n-2)^2 + n^2$
$T(n) = 1 + \sum_{k=1}^{n}(2k)^2$

$\mathbf{T(n) = 1 + 4\sum_{k=1}^{n} k^2 = 1 + \frac{4 * (n(n+1)(2n+1))}{6} = 1 + \frac{2n(n+1)(2n+1))}{3} = 1 + \frac{4n^3+6n^2+2n}{3}}$ (sum of squares of first n even numbers

<u>Big-Oh Complexity</u>

$\mathbf{O(n^3)}$

CSC6013 - Worksheet for Week 5

3.T(n) = T(n-1) + 1/n   and   T(1) = 1

a. Hint: Go online and find a formula for the sum of the first n terms of the "harmonic series".

<u>Substitute (n-1) –> k = 2</u>

T(n-1) = T((n-1)-1) + $\frac{1}{n-1}$

T(n-1) = T(n-2) + $\frac{1}{n-1}$

T(n) = T(n-2) + $\frac{1}{n-1} + \frac{1}{n}$

<u>Substitute (n-2) –> k = 3</u>

T(n-2) = T((n-2)-1) + $\frac{1}{n-2}$

T(n-2) = T(n-3) + $\frac{1}{n-2}$

T(n) = T(n-3) + $\frac{1}{n-2} + \frac{1}{n-1} + \frac{1}{n}$

<u>General Formula</u>

T(n) = T(n-k) + $\frac{1}{n-k+1} + \frac{1}{n-k+2} + \frac{1}{n-k+3} + \ldots + \frac{1}{n-1} + \frac{1}{n}$

<u>Stop Condition</u>

T(1) = 1 → n-k = 1 → **k = n - 1**

T(n) = T(n-(n-1)) + $\frac{1}{n-(n-1)+1} + \frac{1}{n-(n-1)+2} + \frac{1}{n-(n-1)+3} + \ldots + \frac{1}{n-1} + \frac{1}{n}$

T(n) = T(1) + $\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots + \frac{1}{n-1} + \frac{1}{n}$

T(n) = 1 + $\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots + \frac{1}{n-1} + \frac{1}{n}$ **= ln(n) + γ (Sum of first n terms of harmonic series)**

<u>Big-Oh Complexity</u>

- **γ = Euler-Mascheroni constant ≈ 0.58**
  - **Ignore for Big-Oh complexity**

O(ln(n))

CSC6013 - Worksheet for Week 5

Master Method

Compute the complexity of the recursive algorithms based on the recursive equation and stop condition. Show your work, not just your final answer.

4. $T(n) = 2T(n/4) + 1$ and $T(0) = 1$
   a. Be sure to rewrite 1 as n^0.

Variables

a = 2

b = 4

f(n) = n^0

Compare f(n) to n^d

$n^{\log_b a} = n^{\log_4 2} = n^{½}$

$n^0 < n^{½}$

Big-Oh Complexity

$T(n) = O(n^{\log_b a}) = O(n^{1/2})$

5. $T(n) = 2T(n/4) + n^{1/2}$ and $T(0) = 1$
   a. Note that $n^{1/2}$ is the square root of n.

Variables

a = 2

b = 4

f(n) = $n^{1/2}$

Compare f(n) to n^d

$n^{\log_b a} = n^{\log_4 2} = n^{½}$

$n^{1/2} = n^{½}$

Big-Oh Complexity

$T(n) = O(n^{\log_b a}\log(n)) = O(n^{1/2}\log n)$

CSC6013 - Worksheet for Week 5

6. $T(n) = 2T(n/4) + n^2$ and $T(0) = 1$

a. This is similar to the previous one.

Variables

a = 2

b = 4

$f(n) = n^2$

Compare f(n) to n^d

$n^{\log_b a} = n^{\log_4 2} = n^{½}$

$n^2 > n^{½}$

Big-Oh Complexity

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

7. $T(n) = 10T(n/3) + n^2$ and $T(0) = 1$

a. In your answer, round the value of the logarithm to 2 decimal places.
b. Remember that the $\log_b(a)$ is equal to $\log_2(a) / \log_2(b)$.

Variables

a = 10

b = 3

$f(n) = n^2$

Compare f(n) to n^d

$n^{\log_b a} = n^{\log_3 10} = \log_2(10) / \log_2(3) = n^{2.09}$

$n^2 < n^{2.09}$

Big-Oh Complexity

$T(n) = O(n^{\log_b a}) = O(n^{2.09})$

CSC6013 - Worksheet for Week 5

8. $T(n) = 2T(2n/3) + 1$ and $T(0) = 1$

a. In your answer, round the value of the logarithm to 2 decimal places.
b. Be sure to rewrite 1 as n^0.
c. Remember that the $\log_b(a)$ is equal to $\log_2(a) / \log_2(b)$.
d. Hint: rewrite 2n / 3 as n / (3/2)

Variables
a = 2
b = 3/2
$f(n) = n^0$

Compare f(n) to n^d
$n^{\log_b a} = n^{\log_{1.5} 2} = \log_2(2) / \log_2(1.5) = n^{1.71}$

$n^0 < n^{1.71}$

Big-Oh Complexity
$T(n) = O(n^{\log_b a}) = O(n^{1.71})$