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.

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

$T(n) = 2^kT(n-k) + (2^k-1) \quad n-k=0, \; k=n$

$T(n) = 2^nT(0) + (2^n-1) = 2^n + (2^n-1) = 2^{n+1}-1$

$O(2^n)$

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.

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

$T(n) = T(n-2k) + \sum_{i=1}^{k}(n-2i+2)^2 \quad n-2k=0, \; k = n/2$

$T(n) = 1 + \sum_{i=1}^{n/2}(n-2i+2)^2 \approx 1 + \frac{n}{2}(n^2) = \frac{n^3}{2}$

$O(n^3)$

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".

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

$T(n) = T(1) + \sum_{i=2}^{n}\frac{1}{i}$

Harmonic series $\sum_{i=1}^{n}\frac{1}{i}$ grows as $\ln n$ natural logarithm

$O(\ln n)$

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.

$a=2$, $b=4$ $f(n)=1=n^0$

$n^{\log_4(2)} = n^{1/2}$

$n^0 < n^{1/2}$ since $f(n)$ is slower apply rule #1

$O(n^{1/2})$

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

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.

$a=2$, $b=4$, $f(n)=n^{1/2}$

$n^{\log_4(2)} = n^{1/2}$

$n^{1/2} = n^{1/2}$ since $f(n)$ is the same apply rule #2

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

$O(n^{1/2} \log n)$

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

    a. This is similar to the previous one.

$a=2$, $b=4$, $f(n)=n^2$

$n^{\log_4(2)} = n^{1/2}$

$n^2 > n^{1/2}$ since $f(n)$ is faster apply rule #3

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

$O=n^2$

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.

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 $\log_2(a) / \log_2(b)$.

$a=10,\ b=3 \quad f(n)=n^2$

$n^{\log_3(10)} = n^{2.10}$

$n^2 < n^{2.10}$ since f(n) is slower apply case #1

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

$O(n^{2.10})$

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

$2T(n/_{3/2})+1$

   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)

$a=2,\ b=3/2,\ f(n)=1=n^0$

$n^{\log_{3/2}(2)} = n^{1.71}$

$n^0 > n^{1.71}$ since f(n) is slower apply case #1

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

$O(n^{1.71})$