Lab 3 Assignment [Ushna Ijaz- 2019-CE-39]

April 20, 2021

1 Question 4.3.1

Show that the solution of $T(n) = T(n-1) + n$ is $O(n^2)$

Answer:
We know $T(n) \leq cn^2$ for some value of c
$T(n) \leq c(n-1)^2 + n$
$T(n) \leq cn^2 - 2cn + c + n$
Taking c = 1, we get
$n^2 - 2n + 1 + n$
$n^2 - n + 1 \leq n^2$ for $n \geq 1$
$c \geq 1$

2 Question 4.3.2

Question: Show that the solution of $T(n) = T(\lceil n/2 \rceil) + 1$ is $O(\lg n)$

Answer:
$T(n) \leq c \lg(n-2)$
$T(n) \leq c \lg(\lfloor n/2 \rfloor - 2) + 1$
$T(n) = c \lg((n-2)/2) + 1$
$T(n) = c \lg(n-2) - c \lg 2 + 1$
$T(n) \leq c \lg(n-2) - c \lg 2 + 1$
for all values of $c \geq 1$

3 Question 4.3.3

Question: We saw that the solution of $T(n) = 2T(\lfloor n/2 \rfloor) + n$ is $O(n \lg n)$. Show that the solution of this recurrence is also $\Omega(n \lg n)$. Conclude that the solution is $\Theta(n \lg n)$.

Answer:

Our first guess $T(n) \leq cn \lg n$

$T(n) \leq 2c(\lfloor n/2) \lg(\rfloor n/2) + n$

$T(n) \leq cn \lg(n/2) + n$

$T(n) \leq cn \lg n - cn \lg 2 + n$

$T(n) \leq cn \lg n + (1 - c)n$

$T(n) \leq cn \lg n$

for all values of $c \geq \frac{1}{2}$

Next, $T(n) \geq c(n + 2) \lg(n + 2)$

$T(n) \geq 2c(\lfloor n/2 + 2) \lg(\rfloor n/2 + 2) + n$

$T(n) \geq 2c(n/2 - 1 + 2) \lg(n/2 - 1 + 2) + n$

$T(n) \geq 2c(n + 2/2) \lg(n + 2\ 2) + n$

$T(n) = c(n + 2) \lg(n + 2) - c(n + 2) \lg 2 + n$

$T(n) = c(n + 2) \lg(n + 2) + 1(1 - c)n - 2c$

$T(n) = c(n + 2) \lg(n + 2)$

$n \geq 2c/1 - c$ for all values of $0 \leq c \leq 1$

4 Question 4.3.7

Question: Using the master method in Section 4.5, you can show that the solution to the recurrence $T(n) = 4T(n/3) + n$ is $T(n) = \Theta(n^{\log_3 4})$. Show that a substitution proof with the assumption $T(n) \leq cn^{\log_3 4}$ fails. Then show how to subtract off a lower-order term to make the substitution proof work

Answer:

assume $T(n) \leq cn^{\log_3 4}$
guess $T(n) \leq (cn^{\log_3 4}) - n$

$T(n) \leq 4((cn^{\log_3 4}) - n) + n$

$T(n) \leq (cn^{\log_3 4} - 4n + n)$

$T(n) \leq cn^{\log_3 4} - n$

5 Question 4.3.8

Question: Using the master method in Section 4.5, you can show that the solution to the recurrence $T(n) = 4T(n/2) + n^2$ is $T(n) = \Theta(n^2)$. Show that a substitution proof with the assumption $T(n) \leq cn^2$ fails. Then show how to subtract off a lower-order term to make the substitution proof work.

Answer:
$T(n) \leq 4c(n/2)^2 + n^2$

$\leq cn^2 + n^2$
Guess: $T(n) \leq 4c(n/2)^2 - n/2) + n$

$T(n) \leq cn^2 - n$

6 Question 4.3.9

Question: Solve the recurrence $T(n) = 3T(\sqrt{n}) + \log n$ by making a change of variables. Your solution should be asymptotically tight. Do not worry about whether values are integral.

7 Question 4.4.2

Question: Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = T(n/2) + n^2$. Use the substitution method to verify your answer.

Answer: We know that there are $\lg n$ levels
$Sum \sum_{i=0}^{\lg n - 1} (1/4)^i n^2 + 1 = O(n^2)$

Substitution method:
guess: $T(n) \leq cn^2$
$T(n) \leq c(n/2)^2 + n^2$
$T(n) \leq c(n^2)/4 + n^2$
$T(n) \leq (c/4 + 1)n^2$
$T(n) \leq cn^2$
for all values of $c \geq 4/3$

8 Question 4.4.3

Question: Use a reccursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = 2T(n-1) + 1$. Use the substitution method to verify your answer.

Answer: Identifying the depth
depth: $\lg n$
each level adds up and we get n^2 leaves

$T(n) \leq Sum \sum_{i=0}^{\lg n - 1}((2^i n) + (2^{1-i})) + \Theta(n^2)$

$T(n) \leq Sum \sum_{i=0}^{\lg n - 1}(2^i n) + Sum \sum_{i=0}^{\lg n - 1}(2^{1-i}) + \Theta(n^2)$

$\frac{2^(\lg n) - 1}{1} + 2Sum \sum_{i=0}^{\lg n - 1}(1/2)^i + \Theta(n^2)$

$\Theta(n^2) + n + 3$
$\Theta(n^2)$

Substitution method:
Guess: $T(n) \leq cn^2 + n$

$T(n) \leq 4c(n/2)^2 + 2n$

$T(n) \leq cn^2 + 2n$

$T(n) = \Theta(n^2)$

9 Question 4.4.4

Question: Use a reccursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = 2T(n-1) + 1$. Use the substitution method to verify your answer.

Answer: depth = n
each level being $2^i and 2^n$

$T(n) = Sum \sum_{i=0}^{n-1} 2^i + \Theta(2^n)$
Substitution method:
Guess: $T(n) \leq c2^n + n$
$T(n) \leq 2c2^n + (n-1) + 1$
$T(n) = O(2^n)$

10 Question 4.4.5

Question: Use a reccursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = T(n-1)+T(n/2)+n$. Use the substitution method to verify your answer.

Guess: $T(n) \leq c2^n - 4n$

$T(n) \leq c2^{n-1} - 4(n-1) + 2^{n/2}c - \frac{4n}{2} + n$
$T(n) \leq c(2^{n-1} + 2^{n/2}) - 4n - 1$
$T(n) \leq c(2^{n-1} + 2^{n/2}) - 4n - 1$
$T(n) \leq c(2^{n-1}) - 4n$
$T(n) \leq c(2^n) - 4n$
$T(n) = O(2^n)$ for all values of $n \geq 1/2$

Guess: $T(n) \geq cn^2$
$T(n) \geq cn^n - 2cn + 1 + cn^2/4 + n$
$T(n) \geq (2/5)cn^n + (1 + 2c)n + 1$
$T(n) \geq cn^n + (1 + 2)n + 1$
$T(n) \geq cn^2$ for all values of $c \geq 1/3$
$T(n) = O(n^2)$

11 Question 4.4.6

Question: Argue that the solution to the recurrence $T(n) = T(n/3) + T(2n/3) + cn$, where c is a constant, is $\Omega(n \lg n)$ by appealing to the recurrsion tree.

Answer: We know that the cost at each level of the tree is cn. The problem has stated $T(n) = \Omega(n \lg n)$ so now we know for this we find out the lower bound. The shortest path has $log_3 n$ levels . The solution of the recurrence is $cn \log_3 n = \Omega(n \lg n)$

12 Question 4.4.7

Question: Draw the recursion tree for $T(n) = 4T(\lfloor n/2 \rfloor) + cn$, where c is a constant, and provide a tight asymptotic bound on its solution. Verify your answer with the substitution method.

Answer: Tree drawn in word document.

depth: $\lg n$

each level adds up and we get n^2 leaves

$T(n) \leq Sum \sum_{i=0}^{\lg n-1}((2^i cn) + \Theta(n^2)$

$T(n) \leq Sum \sum_{i=0}^{\lg n-1}(2^i) + \Theta(n^2)$

$\frac{2^(\lg n)-1}{1} + \Theta(n^2)$

$\Theta(n^2)$

Guess: $T(n) \leq cn^2 + 2cn$
$T(n) \leq 4cn^2 + 2cn + cn$
$T(n) \leq cn^2 + 2cn$
Guess: $T(n) \geq cn^2 + 2cn$
$T(n) \geq 4cn^2 + 2cn + cn$
$T(n) \geq cn^2 + 2cn$

13 Question 4.5.1

Question: Use the master method to give tight asymptotic bounds for the following recurrences:

a. $T(n) = 2T(n/4) + 1$.
b. $T(n) = 2T(n/4) + \sqrt{n}$.
c. $T(n) = 2T(n/4) + n$.
d. $T(n) = 2T(n/4) + n^2$.
Answer:

a. $\Theta(n^{\log_4 2}) = \Theta(\sqrt{n})$.

b. $\Theta(n^{\log_4 2} \lg n) = \Theta(\sqrt{n} \lg n)$.

c. $\Theta(n)$.

d. $\Theta(n^2)$

14 Question 4.5.2

Question: Professor Caesar wishes to develop a matrix-multiplication algorithm that is asymptotically faster than Strassen's algorithm. His algorithm will use the divide-and-conquer method, dividing each matrix into pieces of size $n/4 \times n/4$, and the divide and combine steps together will take $\Theta(n^2)$ time. He needs to determine how many subproblems his algorithm has to create in order to beat Strassen's algorithm. If his algorithm creates a subproblems, then the recurrence for the running time $T(n)$ becomes $T(n) = aT(n/4) + \Theta(n^2)$. What is the largest integer value of a for which Professor Caesar's algorithm would be asymptotically faster than Strassen's algorithm?

Answer:

We take the case where $a < 16$ and by that we find can tell the algorithm which will be $\Theta(n^2)$.
If we take a=16, in this case our algorithm will be $\Theta(n^2 \log n)$
We know that the running time for Strassen's algorithm is $\Theta(n)^{\lg 7})$ for master to be faster than Strassen, we take $log_4 a \leq \log_2 7$ from this we conclude 7^2 which is equal to 49.
By case 1 of the master theorem, $T(n) = \Theta(n)^{(}\log_4(48)))$

Therefore, the largest integer value of a is 48

15 Question 4.5.3

Question: Use the master method to show that the solution to the binary-search recurrence $T(n) = T(n/2) + \Theta(1)$ is $T(n) = \Theta(\lg n)$.

Answer
a=1
b=2
$T(n) = \Theta(1)$
$T(n) = \Theta(\lg n)$
$T(n) = \Theta(n^{\lg 1})$

16 Question 4.5.4

Question: Can the master method be applied to the recurrence $T(n) = 4T(n/2) + n^2 \lg n$? Why or why not? Give an asymptotic upper bound for this recurrence.

Answer: The master method cannot be applied. It is because we can see that all the 3 cases, one being, taking $a = 4$, $b = 2$, we have $f(n) = n^2 \lg n \neq O(n^{2-\epsilon}) \neq \Omega(n^{2-\epsilon})$.

Prove:
$T(n) \leq 4T(n/2) + n^2 \lg n$
$4c(n/2)^2 \lg^2(n/2) + n^2 \lg n$
$cn^2 \lg(n/2) \lg n - cn^2 \lg(n/2) \lg 2 + n^2 \lg n$
$cn^2 \lg^2 n - cn^2 \lg n \lg 2 - cn^2 \lg(n/2) + n^2 \lg n$
$cn^2 \lg^2 n + (1-c)n^2 \lg n - cn^2 \lg(n/2)(c > 1)$
$T(n) \leq cn^2 \lg^2 n - cn^2 \lg(n/2)$
$T(n) \leq cn^2 \lg^2 n$