CSE 2320 - Homework 4 NAME:__Joshua Abuto____________________________________

Total points: 100 Topics: Recurrences , solved with methods: Master Theorem, Tree

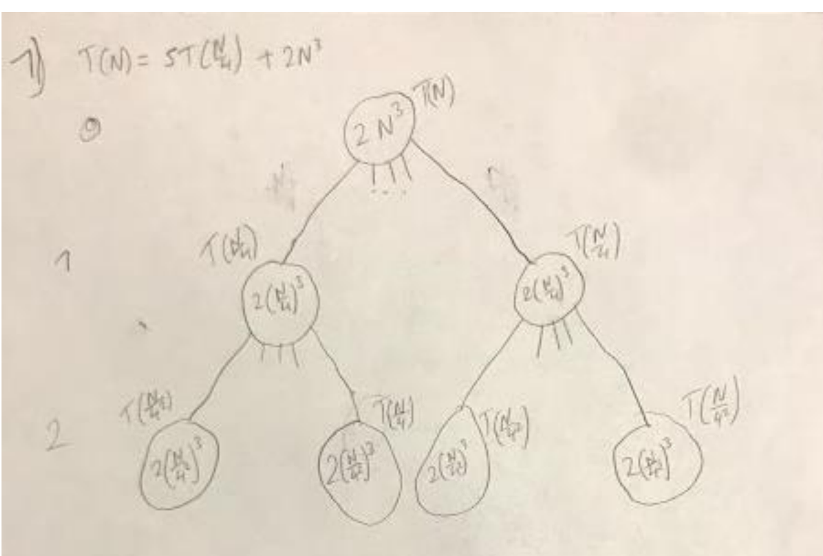
$Convention$: $\lceil\ \rceil$ *means rounded up and* $\lfloor\ \rfloor$ *means rounded down.*

P1. (**23 points**) Use the tree and table method to compute the **Θ** time complexity for $T(N) = 5T(\lfloor N/4 \rfloor) + 2N^3$.
Assume T(0) = 1 and T(1) = 1. Fill in the table below and finish the computations outside of it:

Level	Argument/ Problem size	Cost of one node	Nodes per level	Cost of whole level
0	n	$2n^3$	1	$2n^3$
1	n/4	$2(n/4)^3$	5	$5*2*(n/4)^3= 2n^3*(5/4^3)$
2	$n/4^2$	$2(n/4^2)^3$	5^2	$5^2*2*(n/4^2)^3=2n^3*(5/4^3)^2$
…				
i	$n/4^i$	$2(n/4^i)^3$	5^i	$5^i*2*(n/4^i)^3=2n^3*(5/4^3)^i$
…				
$k=log_4n$ *Leaf level. Write k as a function of N.*	1 $(n/4^k)$	$2 = 2*1 = 2(n/5^k)^3$	5^k	$5^k*2*(n/4^k)^3=2n^3*(5/4^3)^k$

Total tree cost calculation: **$2n^3+2n^3*(5/4^3)+2n^3*(5/4^3)^2+\ldots+2n^3*(5/4^3)^k$** $=2n^3 * \sum_{i=0}^{n}(\frac{5}{4^3})^i = 2n^3 * \left(\frac{1}{1-\frac{5}{4^3}}\right) = 2n^3 * \frac{64}{59}$

T(N) = $\Theta(N^3)$
Draw the tree. Show **levels 0,1,2** and the **leaves level**. Show the problem size T(…) as a label next to the node and inside the node show the local cost (cost of one node) as done in class. For the leaf level and level 2 it suffices to show a few nodes.



P2. (23 **points**) Use the tree and table method to compute the **Θ** time complexity for $T(N) = 4T(N-5) + 7$. Assume T(N) = 7 for all 0≤N≤4. Assume N is a convenient value for your computations.

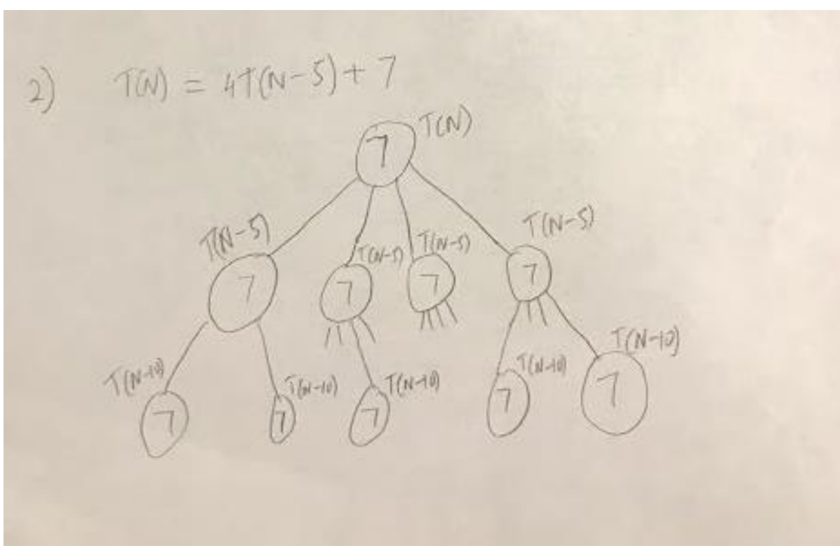
Fill in the table below and finish the computations outside of it:

Level	Argument/ Problem size	Cost of one node	Nodes per level	Cost of whole level
0	N	7	1	1*7
1	n-5	7	4^1	4*7
2	n-5*2	7	4^2	4^2*7
3	n-5*3	7	4^3	4^3*7
4	n-5*4	7	4^4	4^4*7
...				
i	n-5*i	7	4^i	4^i*7
k= *Leaf level. Write k as a function of N.*	n-5*k	7	4^k	4^k*7

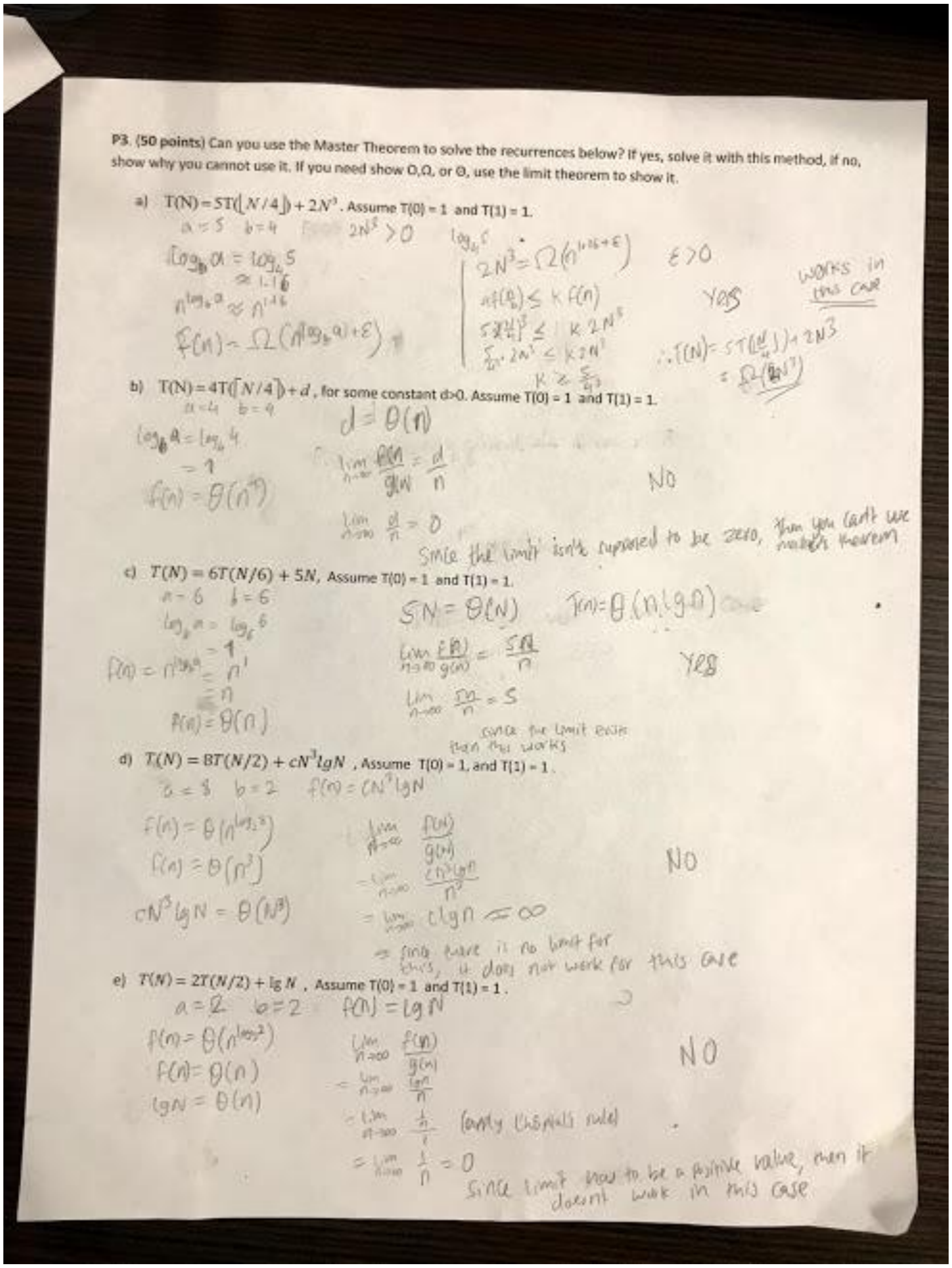
Total tree cost calculation: 7+4*7+4^2*7+...+4^i*7+...+4^k=$7 * \sum_{i=0}^{n} 4^i = 7 * \frac{4^{n+1}-1}{4-1} = \frac{7}{3} * (4^n * 4 - 1)$

T(N) = $\Theta(4^n)$

Draw the tree. Show **levels 0,1,2** and the **leaves level**. Show the problem size T(...) as a label next to the node and inside the node show the local cost (cost of one node) as done in class. For the leaf level and level 2 it suffices to show a few nodes.



P3. (**50 points**) Can you use the Master Theorem to solve the recurrences below? If yes, solve it with this method, if no, show why you cannot use it. If you need show O,Ω, or Θ, use the limit theorem to show it.



P3. (50 points) Can you use the Master Theorem to solve the recurrences below? If yes, solve it with this method, if no, show why you cannot use it. If you need show O,Ω, or Θ, use the limit theorem to show it.

a) $T(N) = 5T(\lfloor N/4 \rfloor) + 2N^3$. Assume T(0) = 1 and T(1) = 1.

$a = 5 \quad b = 4 \quad 2N^3 > 0$

$\log_b a = \log_4 5 \approx 1.16$

$n^{\log_b a} \approx n^{1.16}$

$f(n) = \Omega(n^{\log_b a + \varepsilon})$

$2N^3 = \Omega(n^{1.16+\varepsilon}) \quad \varepsilon > 0$

$af(\frac{n}{b}) \le k f(n)$

$5(\frac{N}{4})^3 \le k 2N^3$

$\frac{5}{4^3} \cdot 2N^3 \le k 2N^3$

$k \ge \frac{5}{4^3}$

Yes

works in this case

$\therefore T(N) = 5T(\lfloor \frac{N}{4} \rfloor) + 2N^3 = \Omega(2N^3)$

b) $T(N) = 4T(\lceil N/4 \rceil) + d$, for some constant d>0. Assume T(0) = 1 and T(1) = 1.

$a = 4 \quad b = 4$

$\log_b a = \log_4 4 = 1$

$f(n) = \Theta(n^1)$

$d = \Theta(n)$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{d}{n}$

$\lim_{n\to\infty} \frac{d}{n} = 0$

NO

Since the limit isn't supposed to be zero, then you can't use master theorem

c) $T(N) = 6T(N/6) + 5N$, Assume T(0) = 1 and T(1) = 1.

$a = 6 \quad b = 6$

$\log_b a = \log_6 6 = 1$

$f(n) = n^{\log_b a} = n^1 = n$

$f(n) = \Theta(n)$

$5N = \Theta(N) \quad T(n) = \Theta(n \lg n)$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{5n}{n}$

$\lim_{n\to\infty} \frac{5n}{n} = 5$

Yes

since the limit exists then this works

d) $T(N) = 8T(N/2) + cN^3 lgN$, Assume T(0) = 1, and T(1) = 1.

$a = 8 \quad b = 2 \quad f(n) = cN^3 \lg N$

$f(n) = \Theta(n^{\log_2 8})$

$f(n) = \Theta(n^3)$

$cN^3 \lg N = \Theta(N^3)$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{cn^3 \lg n}{n^3} = \lim_{n\to\infty} c \lg n = \infty$

NO

⇒ since there is no limit for this, it does not work for this case

e) $T(N) = 2T(N/2) + \lg N$, Assume T(0) = 1 and T(1) = 1.

$a = 2 \quad b = 2 \quad f(n) = \lg N$

$f(n) = \Theta(n^{\log_2 2})$

$f(n) = \Theta(n)$

$\lg N = \Theta(n)$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{\lg n}{n} = \lim_{n\to\infty} \frac{\frac{1}{n}}{1}$ (apply L'Hôpital's rule) $= \lim_{n\to\infty} \frac{1}{n} = 0$

NO

Since limit has to be a positive value, then it doesn't work in this case

P4. (**4 points**) Go to the Wikipedia webpage https://en.wikipedia.org/wiki/Master_theorem_(analysis_of_algorithms) . See section “Inadmissible equations” and list the equation and the reason why it does not satisfy the Master Theorem requirements.

P4. (4 points) Go to the Wikipedia webpage https://en.wikipedia.org/wiki/Master_theorem_(analysis_of_algorithms) . See section “Inadmissible equations” and list the equation and the reason why it does not satisfy the Master Theorem requirements.

$T(n) = 2^n T(\frac{n}{2}) + n^n$ → a is not a constant. The number of subproblems should be fixed

$T(n) = 2T(\frac{n}{2}) + \frac{n}{\log n}$ → Non-polynomial difference between $f(n)$ and $n^{\log_b a}$

$T(n) = 0.5T(\frac{n}{2}) + n$ → $a < 1$ cannot have less than one subproblem

$T(n) = 64T(\frac{n}{8}) - n^2 \log n$ → $f(n)$, which is the combination time is not positive

$T(n) = T(\frac{n}{2}) + n(2 - \cos n)$ → case 3 but regularity violation

Write your answers in a document called **2320_H4.pdf**. It can be hand-written and scanned, but it must be uploaded electronically. Submit just the 2320_H4.pdf.

Remember to include your name at the top.