CSE 2320 - Homework 4 NAME: Apar Pokhrel

Points: 100 Topics: Recurrences, solved with the tree and table method (see Recurrences slides)

P1. (33 pts) 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 multiple of 5. Fill in the table below and finish the computations outside of it. (19 pts)

Level	Argument/ Problem size	Cost of one node	Nodes per level	Cost of whole level
0	N	7	1	$7 \cdot 1$
1	N-5	7	4	$7 \cdot 4$
2	(N-5×2)	7	4^2	$7 \cdot 4^2$
3	(N-5×3)	7	4^3	$7 \cdot 4^3$
i	N-5i	7	4^i	$7 \cdot 4^i$
p= N/5 *Leaf level. Write p as a function of N. (3 pts)*	0 (= N =5p)	7	4^p	$7 \cdot 4^p$

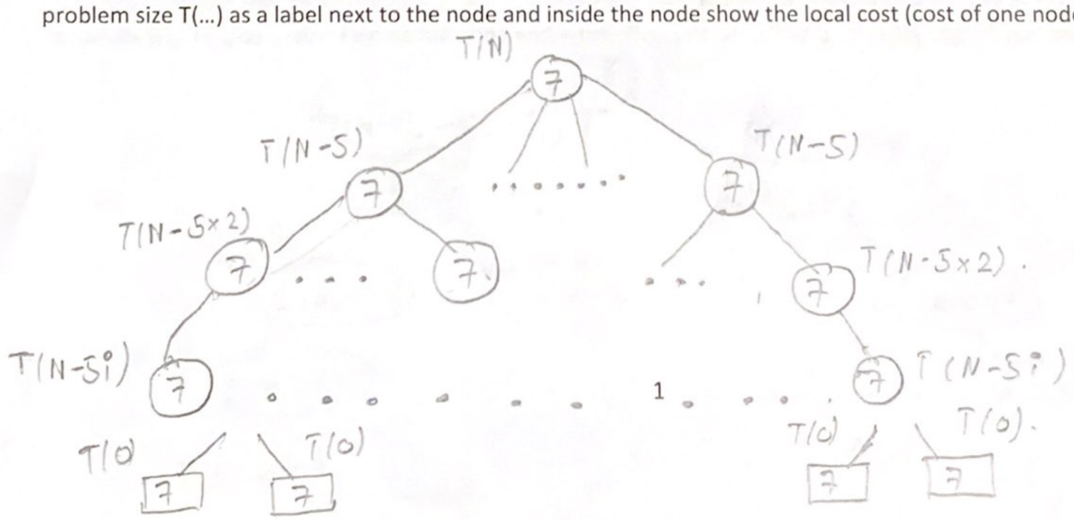
T(0) = >
N-5p = 0
=> p = N/5

b) (8 pts) Total tree cost calculation (give the summation, the general term in the summation and closed form):

$T(N) = 7(1+4+4^2+4^3+\ldots+4^i+\ldots 4^p)$ = $7\left[\frac{4^{\frac{N}{5}+1}-1}{4-1}\right]$ = Θ($4^{N/5}$)

(summation with general term) $= 7\sum_{i=0}^{N/5} 4^i$ (closed form)

C (6pts) Draw the tree (with the root at level 0). Show **the root and its children** and **one node at level i**. 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.



P2. (33 points) Use the tree and table method to compute the **Θ** time complexity for $T(N) = 5T(N/4) + 2N^3$. Assume T(0) = 2 and T(1) = 2. Fill in the table below and finish the computations outside of it. (table: 19 pts total)

Level	Argument/ Problem size	Cost of one node	Nodes per level	Cost of whole level
0	N	$2N^3$	1	$1 \cdot 2N^3$
1	$N/4$	$2(N/4)^3$	5	$5 \cdot 2(N/4)^3$
2	$N/4^2$	$2(N/4^2)^3$	5^2	$5^2 \cdot 2(N/4^2)^3$
3	$N/4^3$	$2(N/4^3)^3$	5^3	$5^3 \cdot 2(N/4^3)^3$
i	$N/4^i$	$2(N/4^i)^3$	5^i	$5^i \cdot 2(N/4^i)^3$
p= $\log_4 N$ *Leaf level. Write p as a function of N. (3 pts)*	1 $(= N/4^p)$	$2(N/4^p)^3$	5^p	$5^p \cdot 2(N/4^p)^3$ $= 2N^3(5/4^3)^p$

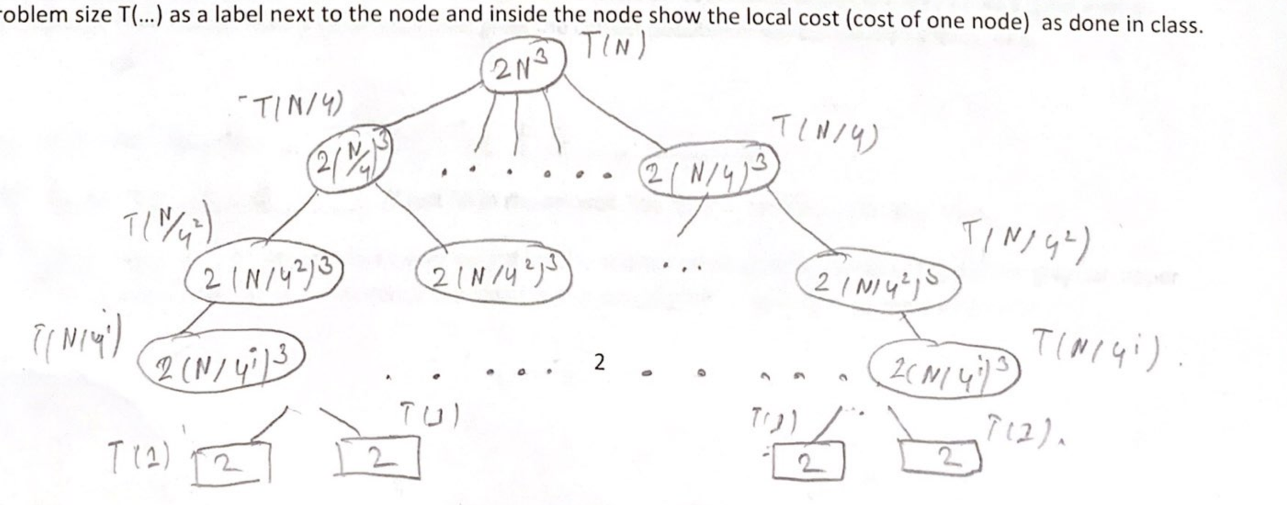
$T(1) \Rightarrow$
$N/4^p = 1 \Rightarrow$
$p = \log_4 N$

b) (8pts) Total tree cost calculation (give the summation, the general term in the summation and closed form):

$T(N) = \sum_{i=0}^{\log_4 N} 2N^3\left(\frac{5}{4^3}\right)^i = 2N^3 \sum_{i=0}^{\log_4 N}\left(\frac{5}{4^3}\right)^i$ = $2N^3 \frac{1-(5/64)^{1+\log_4 N}}{1-(5/64)}$ = Θ(N^3)

(summation with general term) (closed form)

c) (6pts) Draw the tree (with the root at level 0). Show **the root and its children** and **one node at level i**. 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.



P3. (27 points) Given the code below, answer the following questions. Assume the code runs and it produces what it's supposed to. Do not worry about what it calculates.

```
int someFct(int N){
    if (N <= 2) {return 0;}
    if ( N%2==0 ) {return (5+someFct(N/2));}
    else {return (4*someFct(N-3));}
}
```

a) (5)Show tree of fct calls for N = 7?

someFct(N=7)

|

4x someFct(4)

|

5 + somefct(2)

|

return 0.

b) **(9 pts)** Give the recurrence formula for the above code.

(2pts) Base case(s) formula:

$T(N) = 0 \quad ; \text{ if } N \leq 2$

(7 pts) Recursive case recurrence formula as derived directly from the code. You should get a somewhat weird recurrence. Do NOT try to solve it. Hint: how do you write the function expression for absolute value of any real number x (|x| = ...)?

$$
\begin{aligned}
T(N) &= 0 && ; \text{ if } N <= 2. \\
&= 5 + T(N/2) && ; N \text{ is even.} \\
&= 4 \times T(N-3) && ; N \text{ is odd.}
\end{aligned}
$$

c) Give a recurrence that can be used as a *lower bound* for the original recurrence. Note that T(N) = c also gives a lower bound but that is too loose. Give a recurrence that gives the closest possible lower bound, not a loose one.

(3 pts) Recurrence: $T_{lower}(N) =$ $T(N/2) + c.$

(2 pts) $T_{lower}(N) = \Theta($ $\lg N$ $)$ // just fill in the answer. You do not need to show your work.

d) Give a recurrence that can be used as an *upper bound* for the original recurrence. Note that T(N) = c also gives an upper bound but that is too loose. Give a recurrence that gives the closest possible upper bound, not a loose one.

(3 pts) Recurrence: $T_{upper}(N) =$ $T(N-3) + c$

(2 pts) $T_{upper}(N) = \Theta($ N $)$ // just fill in the answer. You do not need to show your work.

e) (3pts) Based on the calculations you did so far, fill in (if you can) the answers below about the original recurrence (for the given code)

$T(N) = \Omega($ $\lg N$ $)$ $T(N) = \Theta($ $)$ $T(N) = O($ N $)$

~~You need to solve: $\sum_{i=0}^{n} 9^i (\frac{n}{8^i})^4$~~

couldn't determine

P4. (7 points)

(2pts) Step a) Rewrite the expression below in such a way that i shows in only one place as the exponent (i.e. a term of a geometric progression)

$9^i (\frac{n}{8^i})^4 = n^4 \left(\frac{9}{8^4}\right)^i$

(5pts) Step b) Use step a) to rewrite the original sum as a sum of geometric series and solve it (give the closed form for it and the Theta.

$\sum_{i=0}^{n} 9^i (\frac{n}{8^i})^4 =$

Closed form: $n^4 \left(\frac{1 - (9/4096)^{1+n}}{1 - (9/4096)} \right)$

$\Theta($ n^4 $)$

Note:

Closed form $< n^4 \cdot \frac{1}{1 - (9/4096)} =$

$\Theta(n^4) = T(n) = O(n^4)$.

But $T(n) = \Omega(n^4) \Rightarrow$

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

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.