CSC 206 Algorithms and Paradigms
CSC 140 Advanced Algorithm Design and Analysis
Spring 2022
Assignment 2: Recurrences
Solutions

Solve **all** the following recurrences using the Master Method (if possible) and the **first five** of them using the recursion tree method:

1. $T(n) = T(\frac{3n}{4}) + c$

\# of leaves = $n^{\log 1}$ (base 4/3, but the base does not matter here)
= n^0 = 1 (same order as c) → Case 2
T(n) = θ(logn)

2. $T(n) = T(\frac{n}{4}) + cn$

\# of leaves = $n^{\log 1}$ (base 4, but the base does not matter here)
= n^0 = 1
cn has a higher order than the number of leaves → Case 3
Regularity condition: $af(n/b) \le k\, f(n)$
$cn/4 \le k\, cn$
Satisfied for k = ¼ < 1
T(n) = θ(n)

3. $T(n) = 5T(\frac{n}{2}) + n^2$

\# of leaves = $n^{\log_2 5} > n^{\log_2 4} = n^2$ (for large n) → Case 1
T(n) = θ($n^{\log_2 5}$)

4. $T(n) = 8T(\frac{n}{2}) + n^3$

\# of leaves = $n^{\log_2 8} = n^3$ (same order as f(n)) → Case 2
T(n) = θ($n^3 \log n$)

5. $T(n) = 8T(\frac{n}{3}) + n^2$

\# of leaves = $n^{\log_3 8} < n^{\log_3 9} = n^2$ (for large n) → Case 3
Regularity condition: $af(n/b) \le k\, f(n)$
$8\,(n/3)^2 \le k\, n^2$
$8\, n^2 / 9 \le k\, n^2$
Satisfied for k = 8/9 < 1
T(n) = θ(n^2)

6. $T(n) = 7T(\frac{n}{6}) + n\log n$

# of leaves = $n^{\log_6 7} > n\log n$ (for large n)

The number of leaves is polynomially larger than nlogn, because $n^{\log_6 7}$ has a polynomial degree greater than 1, while nlogn has a polynomial degree of 1. As proved in class, even the smallest polynomial is larger than logn.

T(n) = θ($n^{\log_6 7}$)

7. $T(n) = 4T(\frac{n}{2}) + n^2 \log^3 n$

# of leaves = $n^{\log_2 4} = n^2 < n^2 \log^3 n$ (for large n)

However, they are both of the same polynomial degree of 2. So, we cannot apply the Master Theorem to this case.

8. $T(n) = 5T(\frac{n}{6}) + n\log n$

# of leaves = $n^{\log_6 5} < n^1 < n\log n$ (for large n)

nlogn is polynomially larger than the number of leaves, because it has a polynomial degree of 1, while $n^{\log_6 5}$ has a polynomial degree less than1.

T(n) = θ(nlogn)

CSC 206 Algorithms and Paradigms

Solutions to Second Assignment

Recursion Tree Method

(1) $T(n) = T\left(\frac{3n}{4}\right) + c$

Height $H = \log_{4/3} n$

$L = 1^H = 1$

Base cost $= c1 \times 1 = c1$

Recursive cost $= c \log_{4/3} n$

Total cost $= c1 + c\log_{4/3} n$

$= \Theta(\log n)$

Level 0 $\quad c$

Level 1 $\quad c$

Level 2 $\quad c$

$\vdots$

Level H $\quad c1$

same cost at all recursive levels

(2) $T(n) = T\left(\frac{n}{4}\right) + cn$

$H = \log_4 n$

$L = 1^H = 1$

Base cost $= c1 \times 1 = c1$

Level 0 $\quad cn$

Level 1 $\quad cn/4$

Level 2 $\quad cn/16$

$\vdots$

Level H $\quad c1$

Rec. cost is not the same at all levels. So, we need a summation

Cost at level $i = (cn/4^i)$

Rec. cost $= \sum_{i=0}^{H-1} cn\left(\frac{1}{4}\right)^i$

$$= cn\left[\frac{\left(\frac{1}{4}\right)^{\log_4 n} - 1}{(1/4) - 1}\right]$$

$$= \frac{4cn}{3}\left[1 - \frac{1}{n^{\log_4 4}}\right] = \frac{4cn}{3} - \frac{4c}{3}$$

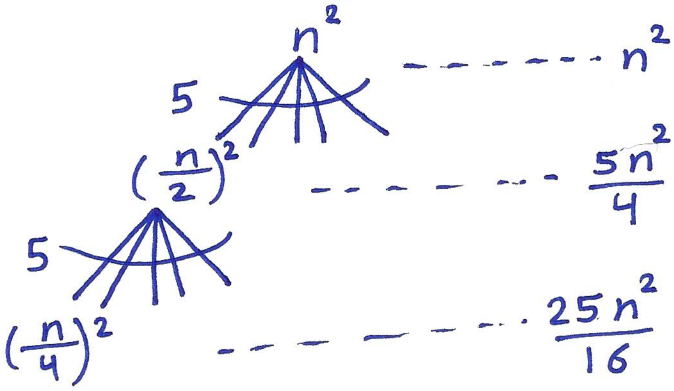
Total cost $= c1 + 4cn/3 - 4c/3 = \Theta(n)$

(3) $T(n) = 5T(\frac{n}{2}) + n^2$

$H = \log_2^n$

$L = 5^H = 5^{\log_2^n} = n^{\log_2^5}$

Base cost $= c1 . n^{\log_2^5}$



Cost is not the same at all levels. So, a summation is needed.

Cost at level $i = \frac{5^i}{4^i} n^2$

Rec. cost $= \sum_{i=0}^{H-1} (\frac{5}{4})^i n^2$

$$= n^2 \left[\frac{(5/4)^{\log_2^n} - 1}{(5/4) - 1} \right]$$

$$= 4n^2 \left[\frac{5^{\log_2^n}}{4^{\log_2^n}} - 1 \right] = 4n^2 \left[\frac{n^{\log_2^5}}{n^2} - 1 \right] = 4n^{\log_2^5} - 4n^2$$

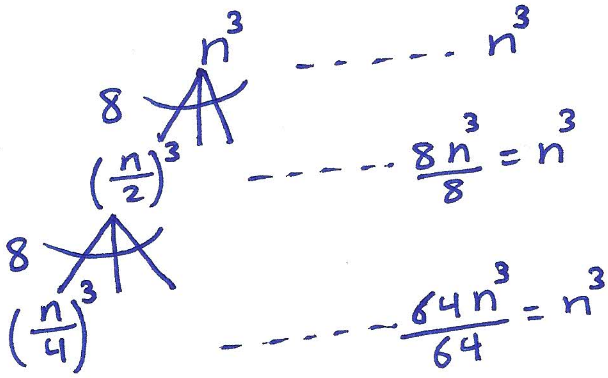
Total cost $= n^{\log_2^5} + 4n^{\log_2^5} - 4n^2 = \Theta(n^{\log_2^5})$

(4) $T(n) = 8T(\frac{n}{2}) + n^3$

$H = \log_2^n$

$L = 8^H = 8^{\log_2^n} = n^{\log_2^8} = n^3$

Base cost $= c1\, n^3$

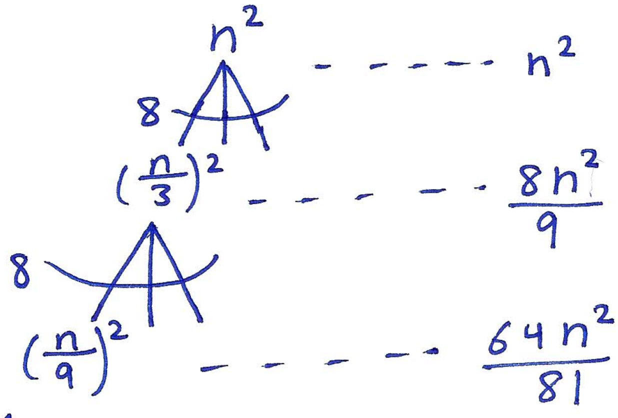


Cost is the same at all recursive levels. So, we simply multiply by H (no summation is needed)

Rec. cost $= n^3 H = n^3 \log_2^n$

Total cost $= c1\, n^3 + n^3 \log_2 n = \Theta(n^3 \log n)$

(5) $T(n) = 8T\left(\frac{n}{3}\right) + n^2$



$H = \log_3 n$

$L = 8^H = 8^{\log_3 n} = n^{\log_3 8}$

Base cost $= c1\, n^{\log_3 8}$

Cost is not the same at all levels. So, a summation is needed.

Cost at level $i = \frac{8^i}{9^i} n^2$

$$\text{Rec. cost} = \sum_{i=0}^{H-1} \left(\frac{8}{9}\right)^i n^2$$

$$= n^2 \left[\frac{\left(\frac{8}{9}\right)^{\log_3 n} - 1}{(8/9) - 1} \right]$$

$$= 9n^2 \left[1 - n^{\log_3 8} / n^{\log_3 9} \right]$$

$$= 9n^2 - 9n^{\log_3 8}$$

Total cost $= c1\, n^{\log_3 8} + 9n^2 - 9n^{\log_3 8} = \theta(n^2)$

Note that n^2 has a higher polynomial degree than $n^{\log_3 8}$