German University in Cairo
Department of Computer Science
Dr. Nourhan Ehab

CSEN 703 Analysis and Design of Algorithms, Winter Term 2022
Practice Assignment 4

Exercise 4-1

Use the master theorem to get the asymptotic notation of the following recurrences. If you fail to get the case, state the gap in which the recurrence falls.

i. $T(n) = 2T(n/2) + n^3$

Solution:
$n^{\log_b a} = n^{\log_2 2} = n^1$ $f(n) = n^3$
It is clear that, $n^3 = \Omega(n)$
This looks like case 3, but we need to prove the first condition of the case
Choosing $\varepsilon = 0.1$
$n^3 = \Omega(n^{1.1})$
Now checking the regularity condition, $af(n/b) \leq cf(n)$

$2(\frac{n}{2})^3 \leq cn^3$

$\frac{2}{8}n^3 \leq cn^3$

dividing by n^3
$\frac{1}{4} \leq c$
Therefore, $\frac{1}{4} \leq c < 1$
Hence, case 3 is proved. So, $T(n) = \Theta(n^3)$

ii. $T(n) = 16T(n/4) + n^2$

Solution:
$n^{\log_b a} = n^{\log_4 16} = n^2$
$f(n) = n^2$
$f(n) = \Theta(n^{\log_b a})$
We are in case 2, therefore
$T(n) = n^2 \lg(n)$

iii. $T(n) = 8T(n/2) + n^2$

Solution:
$n^{\log_b a} = n^{\log_2 8} = n^3$
$f(n) = n^2$
We can see that, $n^2 = O(n^3)$
This looks like case 1, but we need to prove the condition of the case
Choosing $\varepsilon = 0.1$
$n^2 = O(n^{2.9})$
Case 1 is proved. So, $T(n) = \Theta(n^3)$

iv. $T(n) = 5T(n/5) + \frac{n}{\lg(n)}$

Solution:

$n^{\log_b a} = n^{\log_5 5} = n^1$
$f(n) = \frac{n}{\lg(n)}$
It is clear that, $\frac{n}{\lg(n)} = O(n)$
This looks like case 1,
$=> \frac{n}{\lg(n)} = O(n^{1-\varepsilon})$
$=> \frac{n}{\lg(n)} \leq C.n^{1-\varepsilon}$
$=> \lg(n) \geq C.n^{\varepsilon}$

Therefore, there does not exist an epsilon that will satisfy this since all polynomials grow faster than logarithms. We failed to prove case 1 and we fell into a gap between case 1 and 2

v. $T(n) = 5T(n/5) + n\lg(n)$

Solution:
$n^{\log_b a} = n^{\log_5 5} = n^1$
$f(n) = n\lg(n)$
It follows that , $n\lg(n) = \Omega(n)$
This looks like case 3.
$=> n\lg(n) = \Omega(n^{1+\varepsilon})$
$=> n\lg(n) \geq C.n^{1+\varepsilon}$
$=> \lg(n) \geq C.n^{\varepsilon}$

Therefore, there does not exist an epsilon that will satisfy this since all polynomials grow faster than logarithms. we failed to prove case 3 and we fell in a gap between case 2 and 3

Exercise 4-2

Solve the following recurrences using the master theorem. Assume that for very small values of n that T(n) = $\Theta(1)$

i. $T(n) = 4T(n/2) + n^2\sqrt{n}$

Solution:
$n^{\log_b a} = n^{\log_2 4} = n^2$
$f(n) = n^2\sqrt{n}$
It is clear that, $n^2\sqrt{n} = \Omega(n^2)$
This looks like case 3, but we need to prove the first condition of the case
Choosing $\varepsilon = 0.5$
$n^2\sqrt{n} = \Omega(n^2\sqrt{n})$
Now checking the regularity condition, $af(n/b) \leq cf(n)$

$4(\frac{n}{2})^{2.5} \leq cn^{2.5}$

$\frac{1}{\sqrt{2}}n^{2.5} \leq cn^{2.5}$

dividing by $n^{2.5}$
$\frac{1}{\sqrt{2}} \leq c$
Therefore, $\frac{1}{\sqrt{2}} \leq c < 1$
We can choose any value for c e.g. $c = 0.8$
Hence, case 3 is proved. So, $T(n) = \Theta(n^2\sqrt{n})$

ii. $T(n) = 3T(n/2) + n\lg(n)$

Solution:

We can see that, $n\lg(n) = O(n^{1.58}))$
This looks like case 1, but we need to prove the condition of the case
Choosing $\varepsilon = 0.1$

$n \lg(n) = O(n^{1.48})$
Case 1 is proved. So, $T(n) = \Theta(n^{\log_2 3})$

Exercise 4-3 From CLRS (©MIT Press 2001)

The recurrence $T(n) = 7T(n/2) + n^2$ describes the running time of an algorithm A. A competing algorithm A' has a running time of $T'(n) = aT'(n/4) + n^2$. What is the largest integer value for a such that A' is asymptotically faster than A.

Solution:

We start by determining the complexity for $T(n)$. We proceed in the usual way:

$$T(n) = 7 \cdot T\left(\frac{n}{2}\right) + n^2$$

$$a = 7,\ b = 2,\ f(n) = n^2$$

$$\log_2 7 = 2.807354922\ldots \approx 2.8$$

$$n^2 = O(n^{2.8-\epsilon})$$

for $\epsilon = 0.1$ for example, Therefore

$$T(n) = \Theta\left(n^{\lg 7}\right)$$

Now we have a reference to which we can compare $T'(n)$. We note that:

$$f'(n) = n^2 = \begin{cases} O\left(n^{\log_4 a - \epsilon}\right) & \log_4 a > 2 \\ \Theta\left(n^{\log_4 a}\right) & \log_4 a = 2 \\ \Omega\left(n^{\log_4 a + \epsilon}\right) & \log_4 a < 2 \end{cases}$$

$f(n)$ is regular when $a\ f(\frac{n}{4}) = \frac{a}{4^2} \cdot n^2 \leq 1 \cdot n^2$. This will be the case for $a \leq 16$. Assume for a moment that $f(n)$ *is* regular. Then:

$$T'(n) = \begin{cases} \Theta\left(n^{\log_4 a}\right) & \log_4 a > 2 \\ \Theta\left(n^{\log_4 a} \cdot \lg n\right) & \log_4 a = 2 \\ \Theta\left(n^2\right) & \log_4 a < 2 \end{cases}$$

Note that $\log_4 a = 2 \Longrightarrow a = 4^2 = 16$ and that $\log_4 a = \lg \sqrt{a}$ (this will ease comparison to $T(n)$). We rewrite the above as:

$$T'(n) = \begin{cases} \Theta\left(n^{\lg \sqrt{a}}\right) & a > 16 \\ \Theta\left(n^2 \lg n\right) & a = 16 \\ \Theta\left(n^2\right) & a < 16 \end{cases}$$

Now we can see that if we use case 3 for $T'(n)$, it *will* be regular because $a < 16$. Moreover, if $T'(n)$ falls into case 2 or 3, it will be faster than $T(n)$. As for case 1, $\Theta\left(n^{\lg \sqrt{a}}\right) < \Theta\left(n^{\lg 7}\right)$ for $\sqrt{a} < 7 \Rightarrow a < 49$.

Exercise 4-4

Consider the following recurrence.

$$\begin{aligned} T(n) &= T(n/2) + 5^{\lfloor \log_5 n \rfloor} \\ T(1) &= \Theta(1) \end{aligned}$$

Can you solve it using the master method? If "yes", solve it; if "no", explain why.

Solution:

Given the recurrence

$$T(n) = T(n/2) + 5^{\lfloor \log_5 n \rfloor} \tag{1}$$

We know that

$$a = 1, \quad b = 2, \quad f(n) = 5^{\lfloor \log_5 n \rfloor}$$

Getting $n^{\log_b a}$,

$$n^{\log_b a} \quad = \quad n^{\log_2 1} \quad = \quad n^0$$

Given that n is power of 5

$$\begin{aligned} f(n) &= 5^{\lfloor \log_5 n \rfloor} \\ &= 5^{\log_5 n} \\ &= n^{\log_5 5} \\ &= n^1 \end{aligned}$$

Given that n is not a power of 5

$$\begin{aligned} f(n) &= 5^{\lfloor \log_5 n \rfloor} \\ &= 5^{\log_5 n - (\log_5 n \% 1)} \\ &= 5^{\log_5 n}.5^{-\log_5 n \% 1} \\ &= n^1.5^{-\log_5 n \% 1} \end{aligned}$$

Consider the following subsitution

$$d \quad = \quad \log_5(n)\%1$$

Therefore the range of values for d is

$$0 \quad < \quad d \quad < \quad 1$$

Now, we know that for 5^{-d}

$$0.2 < \quad 5^{-d} \quad < 1$$

Therefore, $\forall n$ we can say that

$$f(n) = 5^{\lfloor \log_5 n \rfloor} = en^1 \tag{2}$$

where $0.2 < e \leq 1$.

It is known that

$$en^1 \quad = \quad \Omega(n^0)$$

Therefore we can apply case three of the master theorem
Taking, $\varepsilon = 0.2$

$$en^1 = \Omega(n^{0.2}) \tag{3}$$

Checking the regularity condition,

$$\begin{aligned} af(n/b) &\leq cf(n) \\ 5^{\lfloor \log_5(n/2) \rfloor} &\leq c5^{\lfloor \log_5(n) \rfloor} \end{aligned}$$

Now, Consider the case when n = 2m and m is a power of 5

$$\begin{aligned} af(n/b) &\leq cf(n) \\ 5^{\lfloor \log_5(n/2) \rfloor} &\leq c5^{\lfloor \log_5(n) \rfloor} \\ 5^{\lfloor \log_5(m) \rfloor} &\leq c5^{\lfloor \log_5(2m) \rfloor} \\ 5^{\log_5(m)} &\leq c5^{\lfloor \log_5(m) + \log_5(2) \rfloor} \\ 5^{\log_5(m)} &\leq c5^{\log_5(m)} \end{aligned}$$

Therefore $c \geq 1$ which violates our regularity condition. And hence the given recurrence cannot be solved using the master theorem.