Exercises

4.2.1 Use a recursion tree to determine a good asymptotic upper bound on the recurrence

$$T(n) = 3T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n$$

Use the substitution method to verify your answer.

[solution:]

- $T(n) = 3T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n \xrightarrow{\text{simplified form}} 3T\left(\frac{n}{2}\right) + cn$

- Recurrence tree:

Recurrence tree	Cost per level
cn	$cn = \left(\frac{3}{2}\right)^0 cn$
$\frac{cn}{2}$ $\frac{cn}{2}$ $\frac{cn}{2}$	$3\frac{cn}{2} = \left(\frac{3}{2}\right)^1 cn$
$\frac{cn}{4}$ $\frac{cn}{4}$ $\frac{cn}{4}$ $\frac{cn}{4}$ $\frac{cn}{4}$ $\frac{cn}{4}$ $\frac{cn}{4}$ $\frac{cn}{4}$ $\frac{cn}{4}$	$\frac{9}{4}cn = \left(\frac{3}{2}\right)^2 cn$
$\vdots$	

- General form for the cost per level d: $\left(\frac{3}{2}\right)^d cn$

Question:

- $\boxed{\text{How many tree levels ?}}$

size of the input

level 0 : $n = \frac{n}{2^0}$

level 1 : $n/2 = \frac{n}{2^1}$

level 2 : $n/4 = \frac{n}{2^2}$

level 3 : $n/8 = \frac{n}{2^3}$

$\vdots$

level d : $\frac{n}{2^d} \Rightarrow d = \boxed{\log_2 n}$ levels

- Total tree cost : ~~# levels x cost~~

$$\left(\frac{3}{2}\right)^0 cn + \left(\frac{3}{2}\right)^1 cn + \left(\frac{3}{2}\right)^2 cn + \dots + 3^{\boxed{\log_2 n}}$$

$$= \sum_{i=0}^{\log_2 n - 1} \left(\frac{3}{2}\right)^i cn + n^{\log_2 3}$$

$$= \frac{\left(\frac{3}{2}\right)^{\log_2 n} - 1}{\left(\frac{3}{2}\right) - 1} cn + n^{1,5849\dots}$$

Notes:

Geometric Series:

$$\sum_{k=0}^{n} u^k = \frac{u^{n+1} - 1}{u - 1}$$

$$= \frac{\left(\frac{3}{2}\right)^{\log_2 n} - 1}{\underbrace{\frac{3}{2}}_{(1)} - \underbrace{1}_{(2)}} cn + n^{1,5849\ldots}$$

Notes:

$$\log_2 \frac{3}{2} = \log_2 3 - \log_2 2 = \log_2 3 - 1 = 0,5849\ldots$$

$$= \frac{n^{\log_2 \frac{3}{2}} - 1}{\frac{1}{2}} cn + n^{1,5849\ldots}$$

$$= \left(n^{\log_2 3/2} - 1\right) 2cn + n^{\boxed{1,5849\ldots}}$$

$$= \left(n^{\log_2 3/2} - 1\right) 2cn + n^{\boxed{\log_2 3}} \xRightarrow{n \to \infty} n^{\log_2 3}$$

__Hypothesis:__ $T(n) = O\left(n^{\log_2 3}\right)$

__Proof:__

- Hypothesis: $T(n) = O\left(n^{\log_2 3}\right) = \boxed{d n^{\log_2 3}}$
- $T(n) = 3T\left(\frac{n}{2}\right) + cn$

$$\leq 3d\left(\frac{n}{2}\right)^{\log_2 3} + cn$$

$$= 3d\frac{n^{\log_2 3}}{2^{\log_2 3}} + cn$$

$$= \cancel{3} d \frac{n^{\log_2 3}}{\cancel{3}} + cn$$

$$= \cancel{d n^{\log_2 3}} + cn \leq \boxed{\cancel{d n^{\log_2 3}}}$$

$$\Leftrightarrow cn \leq 0 \Leftrightarrow c < 0$$

Exercise 4.2.2 Argue that the solution to the recurrence $T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + cn$, where c is a constant, is $\Omega(n \lg n)$ by appealing to a recursion tree

[resolution:]

Recurrence: $T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + cn$ ~~Simplified form $T(n) = T($~~

	Cost per level
cn	cn
$c\left(\frac{n}{3}\right) \quad c\left(\frac{2n}{3}\right)$	cn
$c\left(\frac{n}{9}\right) \quad c\left(\frac{2n}{9}\right) \quad c\left(\frac{2n}{9}\right) \quad c\left(\frac{4n}{9}\right)$	cn
⋮	

General cost at level d: cn

Observation:

- Each recurrence calls $T\left(\frac{n}{3}\right)$ and $T\left(\frac{2n}{3}\right)$
- This leads to an unbalanced tree
- The longest path will be when we follow the $T\left(\frac{2n}{3}\right)$ decompositions. The longest path is the one we should follow for the worst-case behaviour (O-notation)
- The shortest path will be when we follow the $T\left(\frac{n}{3}\right)$ decompositions. The shortest path is the one we should follow for the best-case behaviour (Ω-notation)

Lets see then what happens for $T\left(\frac{n}{3}\right)$ decompositions:

<u>Size of the input:</u>

level 0: $\left(\frac{1}{3}\right)^0 n$

level 1: $\left(\frac{1}{3}\right)^1 n$

level 2: $\left(\frac{1}{3}\right)^2 n$

level 3: $\left(\frac{1}{3}\right)^3 n$

⋮

level d: $\left(\frac{1}{3}\right)^d n = \frac{n}{\left(\frac{3}{1}\right)^d} = \frac{n}{3^d} \Rightarrow \boxed{d = \log_3 n}$ levels

Total tree cost: # levels x cost Per Level

$= \log_3 n \times cn \Rightarrow \Omega(n \log n)$

4.2.4 Use a recursion tree to give an asymptotically tight solution to the recurrence

$$T(n) = T(n-a) + T(a) + cn$$

where $a \geq 1$ and $c > 0$ are constants

[resolution:]

- Recurrence: $T(n) = T(n-a) + T(a) + cn$

Recursion tree	Cost Per Level
cn	cn
$c(n-a)$ ca	$cn - ca + ca = cn$
$c(n-2a)$ ca $c(a-a)$ ca …	$cn - 2ca + ca + 0 + ca = cn$

- General cost for level d : cn

- $\boxed{\text{Question:}}$ $\boxed{\text{How many tree levels ?}}$

Size of the input:

level 0: n

level 1: $n-a$

level 2: $n-2a$

level 3: $n-3a$

⋮

level d: $n-da$ ⇒ Eventually the input will have size 1, i.e.:

$$n - da = 1 \iff d = \frac{1-n}{-a} = \frac{n-1}{a}$$

- Total Tree Cost = # Levels x Cost Per Level

$$= \frac{n-1}{a} \times cn = \frac{(n-1)cn}{a} = \frac{cn^2 - cn}{a}$$

$$\Rightarrow \lim_{n\to\infty} O(n^2)$$

4.2.5 Use a recursion tree to give an asymptotically tight solution to the recurrence

$$T(n) = T(\alpha n) + T[(1-\alpha)n] + cn$$

where $\underline{\alpha}$ is a constant in the range $0 < \alpha < 1$ and $c > 0$ is also a constant

[resolution:]

- Recurrence:

Cost per Level

$$cn \quad \cdots\cdots \quad cn$$

$$c\alpha n \quad c(1-\alpha)n \quad \cdots\cdots \quad c\alpha n + cn - c\alpha n = cn$$

$$c\alpha^2 n \quad c(1-\alpha)\alpha n \quad \alpha(1-\alpha)n \quad (1-\alpha)(1-\alpha)n \cdots \quad cn$$

- General cost at level $\underline{d}$: $\underline{cn}$
- $\boxed{\text{Question:}}$ How many tree levels ?
  - We need to make some assumptions about α
  - Without loss of generality let $\alpha \geq 1-\alpha$ so that

$$0 < 1-\alpha \leq 1/2$$

$$\frac{1}{2} \leq \alpha < 1$$

(longest path is for $T[(1-\alpha)n]$)

- Size of input:

level 0 : n
level 1 : $(1-\alpha)\, n$
level 2 : $(1-\alpha)^2 n$
level 3 : $(1-\alpha)^3 n$
⋮
level $\underline{d}$: $(1-\alpha)^d n \Longrightarrow$

Eventually the size of the input will be 1, i.e.:

$$(1-\alpha)^d n = 1 \Leftrightarrow \cancel{(1-\alpha)^d = \frac{1}{n}}$$

$$\Leftrightarrow n = \frac{1}{(1-\alpha)^d} \Leftrightarrow n = \left(\frac{1}{1-\alpha}\right)^d \Rightarrow \boxed{d = \log_{\frac{1}{1-\alpha}} n}$$

- Total Tree Cost = # Levels x Cost Per Level

$$= \log_{\frac{1}{1-\alpha}} n \times cn \overset{\lim_{n\to\infty}}{\Longrightarrow} O(n \log n)$$