

Exercises

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

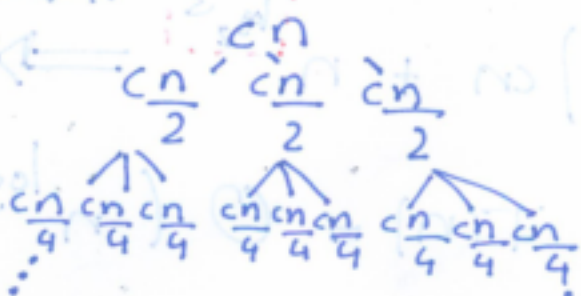
$$T(n) = 3T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n$$

Use the substitution method to verify your answer.

[solution:]

• $T(n) = 3T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n$ Simplified form: $3T\left(\frac{n}{2}\right) + cn$

• Recurrence tree:



Cost per level

$$cn = \left(\frac{3}{2}\right)^0 cn$$

$$3\frac{cn}{2} = \left(\frac{3}{2}\right)^1 cn$$

$$\frac{9}{4}cn = \left(\frac{3}{2}\right)^2 cn$$

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

Question:

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

⋮

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

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

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

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

$$= \frac{\left(\frac{3}{2}\right)^{\log_2 n + 1} - 1}{\left(\frac{3}{2}\right) - 1} cn + n \log_2 n \approx 1.5848 \dots n$$

$\log_2 n$

Notes:

Geometric Series:

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

$$= \frac{\left(\frac{3}{2}\right)^{\log_2 n} - 1}{\frac{3}{2} - 1} cn + n^{1.58495...}$$

14/

$$= \frac{\frac{3}{2}^{\log_2 n} - 1}{\frac{3}{2} - 1} cn + n^{1.58495...}$$

Notes:
 $\log_2 \frac{3}{2} = \log_2 3 - \log_2 2$
 $= \log_2 3 - 1$
 $= 0.58495...$

$$= \frac{n^{\log_2 \frac{3}{2}} - 1}{\frac{3}{2} - 1} cn + n^{1.58495...}$$

$$= (n^{\log_2 \frac{3}{2}} - 1) 2cn + n^{1.58495...}$$

$$= (n^{\log_2 \frac{3}{2}} - 1) 2cn + n^{\log_2 3}$$

Hypothesis: $T(n) = O(n^{\log_2 3})$

Proof:

• Hypothesis: $T(n) = O(n^{\log_2 3}) = dn^{\log_2 3}$

• $T(n) = 3T(\frac{n}{2}) + cn$

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

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

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

$$= \cancel{dn^{\log_2 3}} + cn \leq \cancel{dn^{\log_2 3}}$$

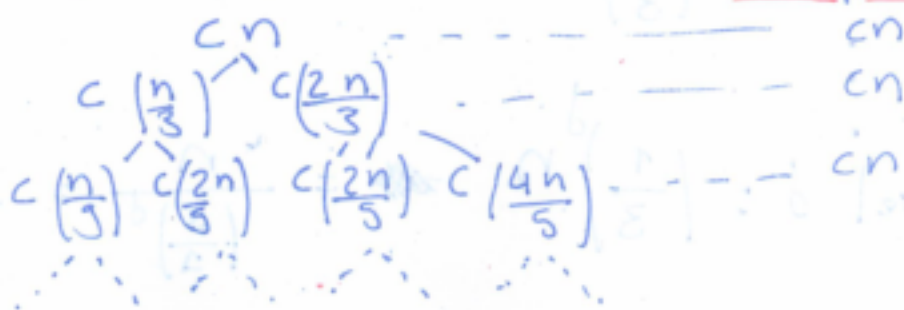
$$\Rightarrow cn \leq 0 \Rightarrow c < 0$$

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~~ ~~$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + cn$~~ Cost per level



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)

Let's see then what happens for $T(\frac{n}{3})$ decompositions:

Size of the input:

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

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

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

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

⋮

level d: $(\frac{1}{3})^d n = \frac{n}{(\frac{3}{1})^d} = \frac{n}{3^d} \Rightarrow \boxed{d = \log_3 n}$
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: $T(n) = T(n-a) + T(a) + cn$ Cost Per Level

$$\begin{array}{ccccccc} & & cn & & & & cn \\ & & \vdots & & & & \vdots \\ & & c(n-a) & & & & cn - ca + ca = cn \\ & & \vdots & & & & \vdots \\ & & c(n-2a) & & & & cn - 2ca + ca + 0 + ca = cn \\ & & \vdots & & & & \vdots \\ & & c(a-a) & & & & ca \end{array}$$

- General cost for level d : cn

• Question: How many tree levels?

Size of the input:

level 0: n

level 1: $n-a$

level 2: $n-2a$

level 3: $n-3a$

\vdots

level d : $n-da \Rightarrow$

Eventually the input will have size 1, i.e.:

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

- Total Tree Cost = #Levels \times Cost Per Level

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

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

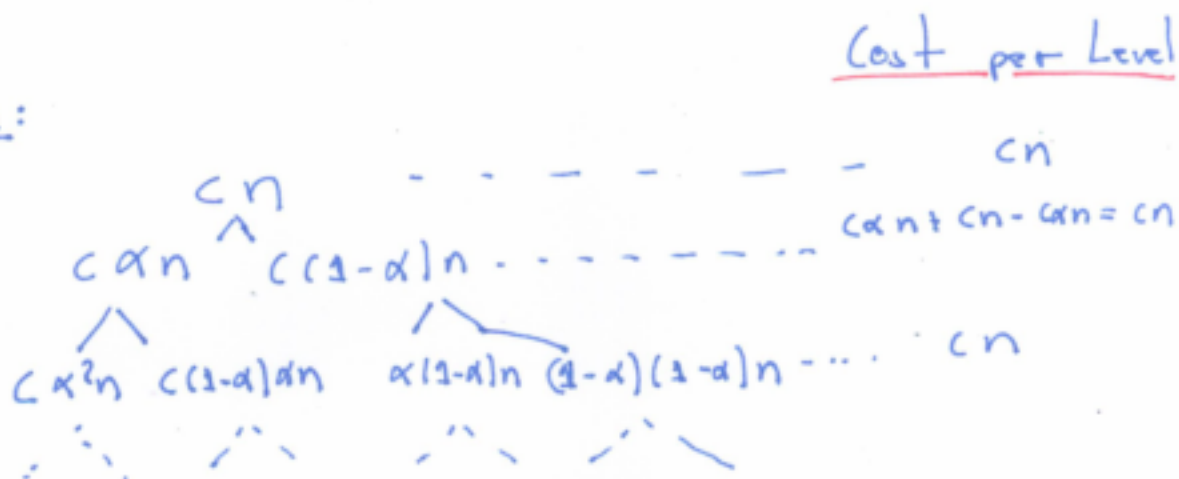
4.2.5 Use a recursion tree to give an asymptotically tight solution to the recurrence

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

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

Resolution:

• Recurrence:



• General cost at level d : cn

• Question: How many tree levels?

- We need to make some assumptions about α
- Without loss of generality let $\alpha \geq 1-\alpha$ so that $0 < 1-\alpha \leq 1/2$
 $\frac{1}{2} \leq \alpha < 1$

• Size of input: (longest path is for $T[(1-\alpha)n]$)

level 0: n
 level 1: $(1-\alpha)n$
 level 2: $(1-\alpha)^2 n$
 level 3: $(1-\alpha)^3 n$
 \vdots
 level d : $(1-\alpha)^d n \Rightarrow$

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

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

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

• Total Tree Cost = #Levels x Cost Per Level

$$= \log_2 n \times cn \stackrel{\lim_{n \rightarrow \infty}}{\Rightarrow} O(n \log n)$$

As $n > 0$... α is a constant ... $\alpha > 0$ is also a constant.

Cost per level

Recurrence



• General cost at level q : C_q

Question: How many tree levels?

We need to work some roughness about α . Without loss of generality let $\alpha \leq 2 - \alpha$ so that $0 < 2 - \alpha \leq 2$.

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

Proof with α for $T(n)$

- Level 0: n
- Level 1: $n/2$
- Level 2: $n/4$
- Level 3: $n/8$
- ...
- Level q : $n/2^q$

Example: the case of the cost

will be $\alpha \cdot n$... $\alpha \cdot n/2^q$... $\alpha \cdot n/2^q$