the recursion-tree method

1. solving recurrences
   - expanding the recurrence into a tree
   - summing the cost at each level
   - applying the substitution method

2. another example
   - using a recursion tree

MCS 360 Lecture 39
Introduction to Data Structures
Jan Verschelde, 22 November 2010

solving recurrences

The substitution method for solving recurrences consists of two steps:

1. Guess the form of the solution.
2. Use mathematical induction to find constants in the form and show that the solution works.

In the previous lecture, the focus was on step 2.

Today we introduce the recursion-tree method to generate a guess for the form of the solution to the recurrence.

the recursion-tree method

1. solving recurrences
   - expanding the recurrence into a tree
   - summing the cost at each level
   - applying the substitution method

2. another example
   - using a recursion tree

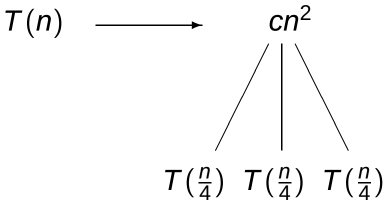
an example

Consider the recurrence relation

$$T(n) = 3T(n/4) + cn^2 \quad \text{for some constant } c.$$

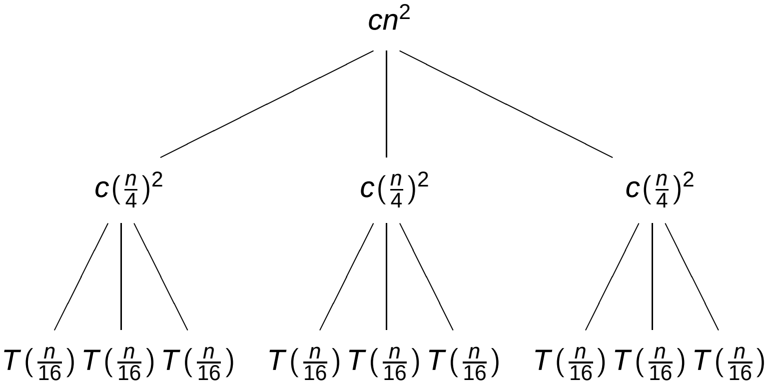
We assume that n is an exact power of 4.

In the recursion-tree method we expand $T(n)$ into a tree:



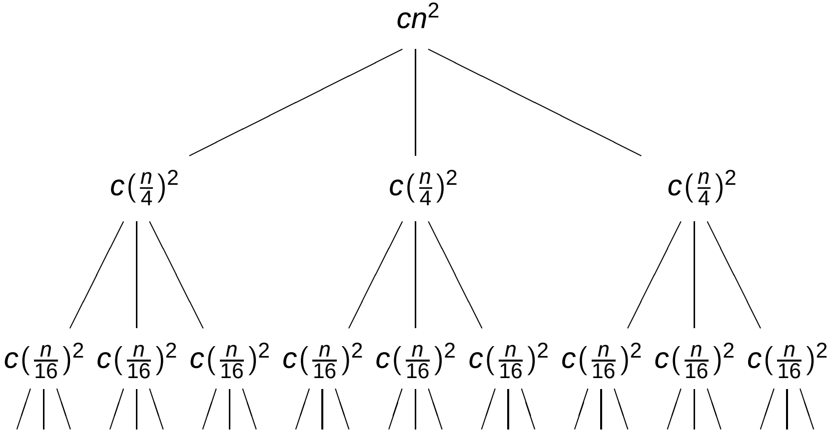
we expand $T(\frac{n}{4})$

Applying $T(n) = 3T(n/4) + cn^2$ to $T(n/4)$ leads to $T(n/4) = 3T(n/16) + c(n/4)^2$, expanding the leaves:



we expand $T(\frac{n}{16})$

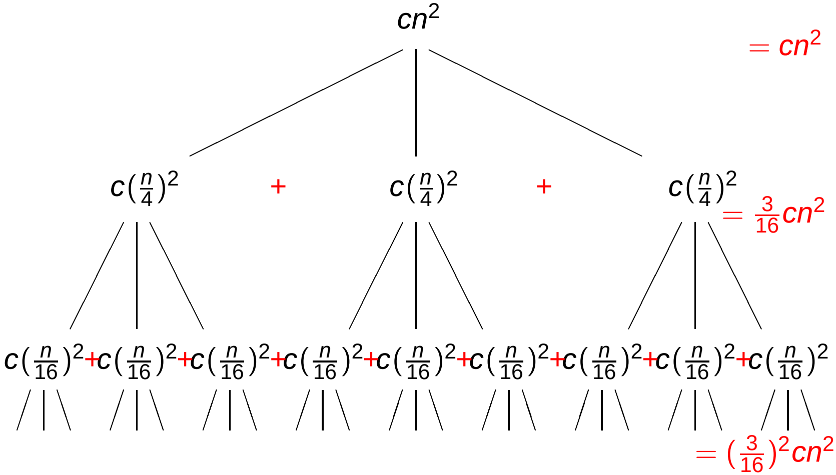
Applying $T(n) = 3T(n/4) + cn^2$ to $T(n/16)$ leads to $T(n/16) = 3T(n/64) + c(n/16)^2$, expanding the leaves:



the recursion-tree method

the cost at each level

We sum the cost at each level of the tree:



adding up the costs

$$
\begin{aligned}
T(n) &= cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \cdots \\
&= cn^2 \left(1 + \frac{3}{16} + \left(\frac{3}{16}\right)^2 + \cdots \right)
\end{aligned}
$$

The $\cdots$ disappear if $n = 16$,
or the tree has depth at least 2 if $n \geq 16 = 4^2$.

For $n = 4^k$, $k = \log_4(n)$, we have:

$$
T(n) = cn^2 \sum_{i=0}^{\log_4(n)} \left(\frac{3}{16}\right)^i .
$$

geometric series

Consider a finite sum first:

$$S_n = 1 + r + r^2 + \cdots + r^n = \sum_{i=0}^{n} r^i.$$

To find an explicit form of the solution we do

$$\begin{array}{rcccccccccccc} rS_n & = & & & r & + & r^2 & + & \cdots & + & r^n & + & r^{n+1} \\ -S_n & = & 1 & + & r & + & r^2 & + & \cdots & + & r^n & & \\ \hline (r-1)S_n & = & -1 & & & & & & & & & + & r^{n+1} \end{array}$$

So the explicit sum is

$$S_n = \frac{r^{n+1} - 1}{r - 1}.$$

applying the geometric sum

Applying

$$S_n = \sum_{i=0}^{n} r^i = \frac{r^{n+1} - 1}{r - 1}$$

to

$$T(n) = cn^2 \sum_{i=0}^{\log_4(n)} \left(\frac{3}{16} \right)^i$$

with $r = \frac{3}{16}$ leads to

$$T(n) = cn^2 \frac{\left(\frac{3}{16}\right)^{\log_4(n)+1} - 1}{\frac{3}{16} - 1}.$$

polishing the result

Instead of $T(n) \leq dn^2$ for some constant d, we have

$$T(n) = cn^2 \frac{\left(\frac{3}{16}\right)^{\log_4(n)+1} - 1}{\frac{3}{16} - 1}.$$

Recall

$$T(n) = cn^2 \sum_{i=0}^{\log_4(n)} \left(\frac{3}{16}\right)^i.$$

To remove the $\log_4(n)$ factor, we consider

$$\begin{aligned} T(n) &\leq cn^2 \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i \\ &= cn^2 \frac{-1}{\frac{3}{16} - 1} \leq dn^2, \text{ for some constant } d. \end{aligned}$$

the recursion-tree method

verifying the guess

Let us see if $T(n) \leq dn^2$ is good for $T(n) = 3T(n/4) + cn^2$.

Applying the substitution method:

$$\begin{aligned}
T(n) &= 3T(n/4) + cn^2 \\
&\leq 3d\left(\frac{n}{4}\right)^2 + cn^2 \\
&= \left(\frac{3}{16}d + c\right) n^2 \\
&= \frac{3}{16}\left(d + \frac{16}{3}c\right) n^2 \\
&\leq \frac{3}{16}(2d)\, n^2, \quad \text{if } d \geq \frac{16}{3}c \\
&\leq dn^2
\end{aligned}$$

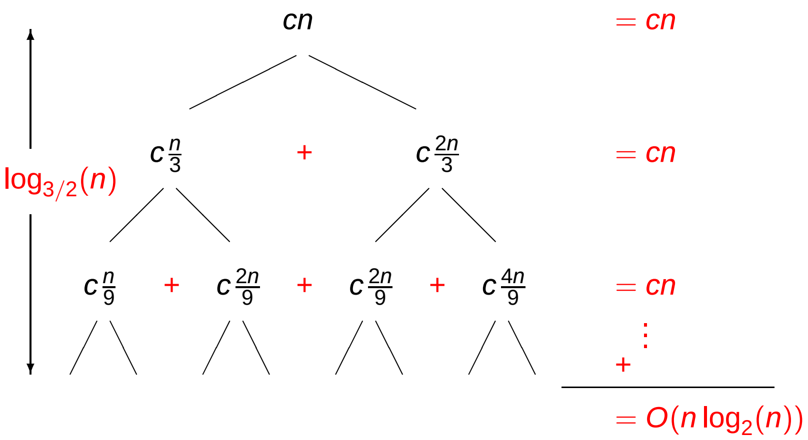
the recursion-tree method

1. solving recurrences
   - expanding the recurrence into a tree
   - summing the cost at each level
   - applying the substitution method

2. another example
   - using a recursion tree

using a recursion tree

Consider $T(n) = T(n/3) + T(2n/3) + cn$.



Summary + Assignments

We covered §4.4 of *Introduction to Algorithms*, 3rd edition by Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson.

Assignments:

1. Consider $T(n) = 3T(n/2) + n$. Use a recursion tree to derive a guess for an asymptotic upper bound for $T(n)$ and verify the guess with the substitution method.
2. Same question as before for $T(n) = T(n/2) + n^2$.
3. Same question as before for $T(n) = 2T(n-1) + 1$.

Last homework collection on Monday 29 November:
#1 of L-30, #1 of L-31, #3 of L-32, #2 of L-33, #1 of L-34.

Final exam on Tuesday 7 December, 8-10AM in TH 216.