Australian National University

COMP3600/6466 Algorithms

Lecture 5

S2 2016

Dr. Hassan Hijazi

Prof. Weifa Liang

Pseudocode Analysis

Iterative

Algorithm 1 My algorithm

```
procedure MyProcedure
    stringlen ← length of string
    i ← patlen
top:
    if i > stringlen then return false
    j ← patlen
loop:
    if string(i) = path(j) then
        j ← j − 1.
        i ← i − 1.
        goto loop.
        close;
    i ← i + max(delta_1(string(i)), delta_2(j)).
    goto top.
```

Recursive

Algorithm 1 My algorithm

```
procedure MyProcedure
    stringlen ← length of string
    i ← patlen
top:
    if i > stringlen then return false
    j ← patlen
loop:
    if string(i) = path(j) then
        j ← j − 1.
        i ← i − 1.
        goto loop.
        close;
    i ← i + max(delta_1(string(i)), delta_2(j)).
    goto top.
```

Recursive Algorithms

Total running time
=
Sum of times in each node of the recursion tree.

Can also be written as a recursion!

For example, the running time $T(n)$ can be expressed:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n \text{ is small,} \\ 2T(\lfloor \frac{n}{2} \rfloor) + \Theta(n) & \text{otherwise.} \end{cases}$$

We usually write:
$T(n) = 2T(\lfloor \frac{n}{2} \rfloor) + \Theta(n)$ assuming that $T(n)$ is constant for small values of n.

Asymptotic Bounds for Recursions

Substitution method:

Guess a bound and use mathematical induction to prove that the guess is correct.

Recursion-tree method:

Convert the recurrence into a tree,

Use this tree to rewrite the function as a sum,

Use techniques of bounding summations to solve the recurrence.

Substitution method

The substitution method consists of two steps:

- Step 1. Guess the form of the solution.
- Step 2. Use mathematical induction to show that the guess is correct.

It can be used to obtain either upper or lower bounds on a recurrence.

A good guess is vital when applying this method.
If the initial guess is wrong, it needs to be adjusted later.

Exercise 5.1

Using the substitution method, prove that

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

Complete the proof by hand

Exercise 5.2

Give an asymptotic upper bound for

$$T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil\right) + 5$$

Complete the proof by hand

Australian National University

Asymptotic Bounds for Recursions

How to make a good guess?

Experience

Recursion Tree

Recursion Tree

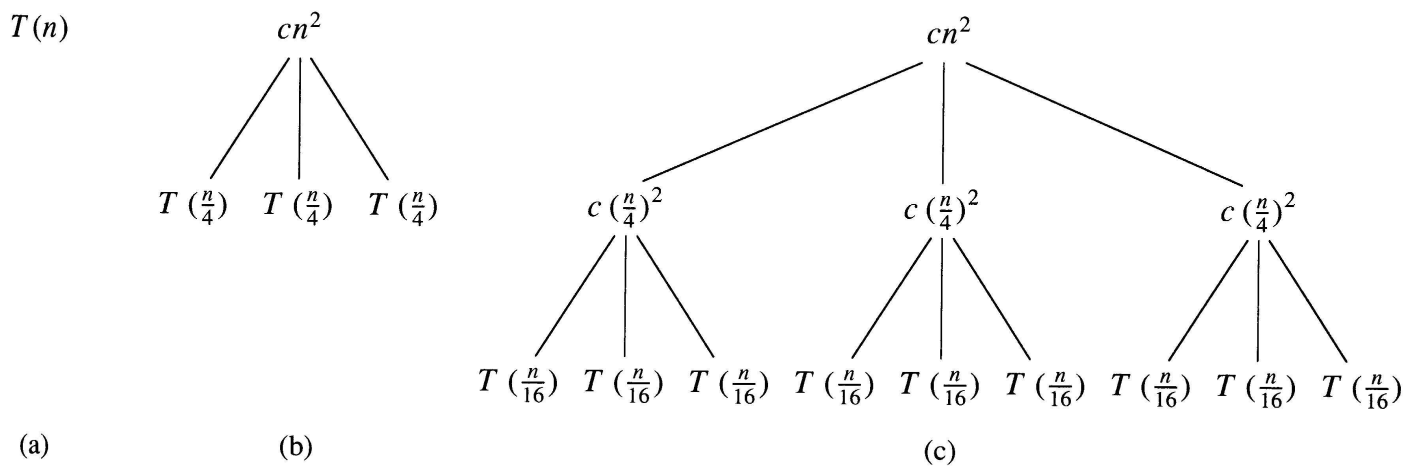
- Also called iteration method
- Can be used to guess or find the solution
- When guessing, we can make simplifying assumptions (e.g., ignore floor and ceiling)
- The goal is to expand the recurrence and express it as a summation

Recursion Tree

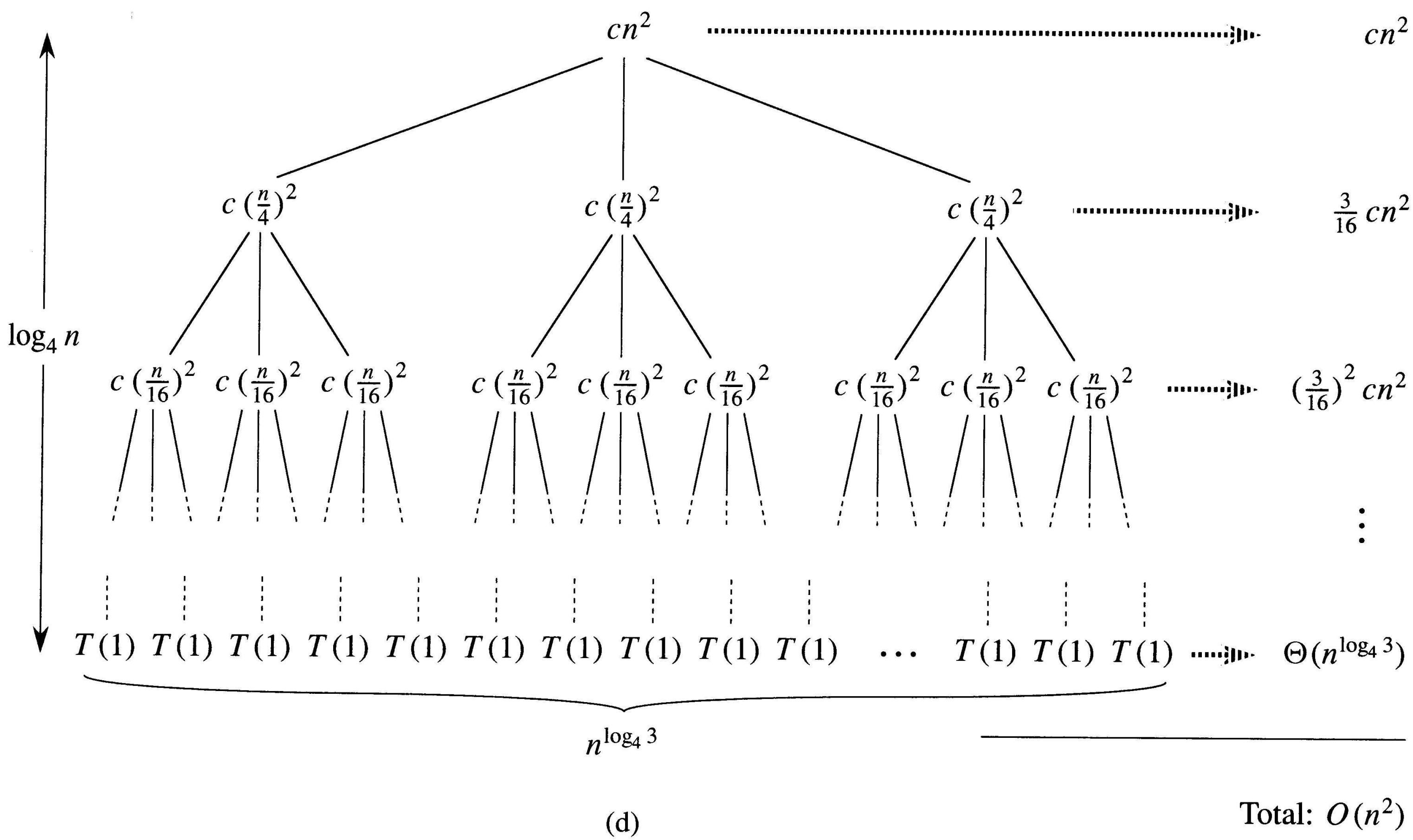
Consider the recurrence

$$T(n) = 3T\left(\left\lfloor \frac{n}{4} \right\rfloor\right) + cn^2$$

Simplification: we assume that n is a power of 4.



Recursion Tree



(d)

Recursion Tree

At depth d the subproblem size is $\frac{n}{4^d}$.

We stop building the tree when we reach subproblem size 1, so when $\frac{n}{4^d} = 1$.

This gives $i = \log_4 n$. Thus, the depth of the tree is $\log_4 n$.

The number of levels is $\log_4 n + 1$.

Recursion Tree

The cost at depth d is $\left(\frac{3}{16}\right)^d cn^2$, except for the bottom level, whose cost is its number of nodes times $T(1)$, that is, $3^{\log_4 n} \cdot T(1) = n^{\log_4 3} \cdot T(1) = \Theta(n^{\log_4 3})$.

This leads to:

$$T(n) = \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) = cn^2 \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{16}\right)^i + \Theta(n^{\log_4 3}).$$

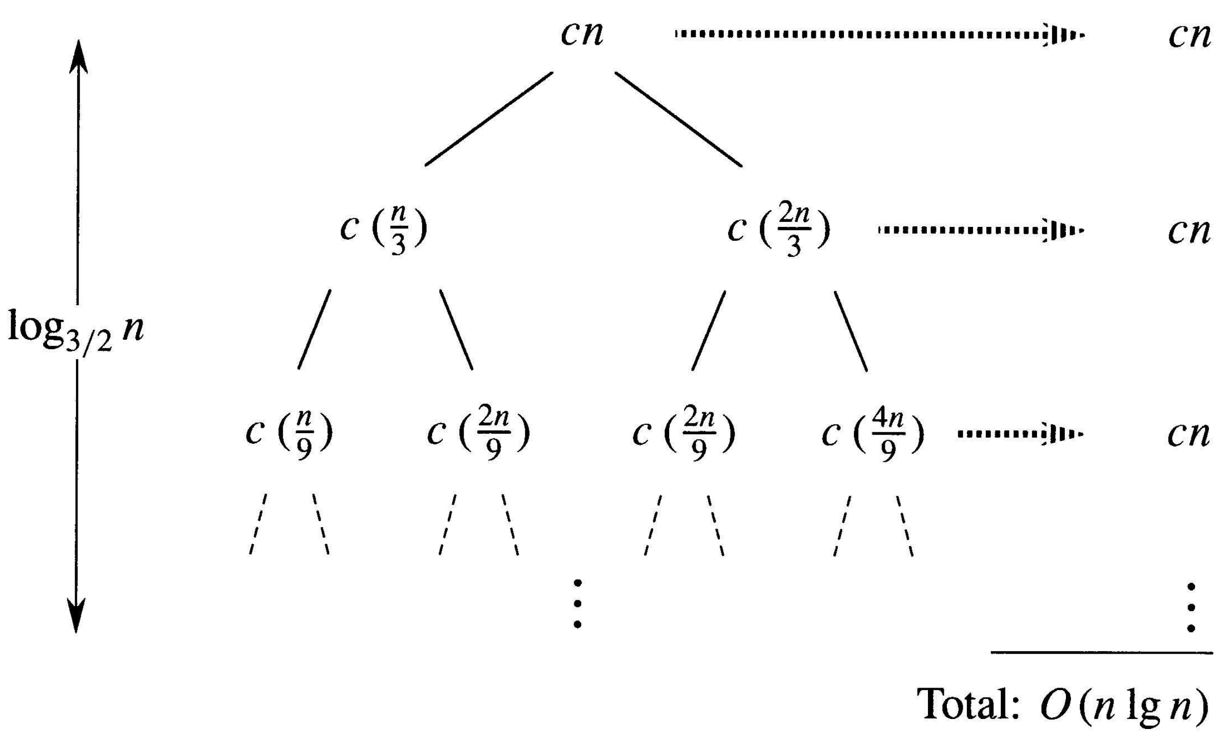
Note that $\displaystyle\sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{16}\right)^i = \Theta(\text{largest term}) = \Theta(1)$ (decreasing geometric sum).

Finally, observe that n^2 grows faster than $n^{\log_4 3}$ as its exponent is larger. Thus,

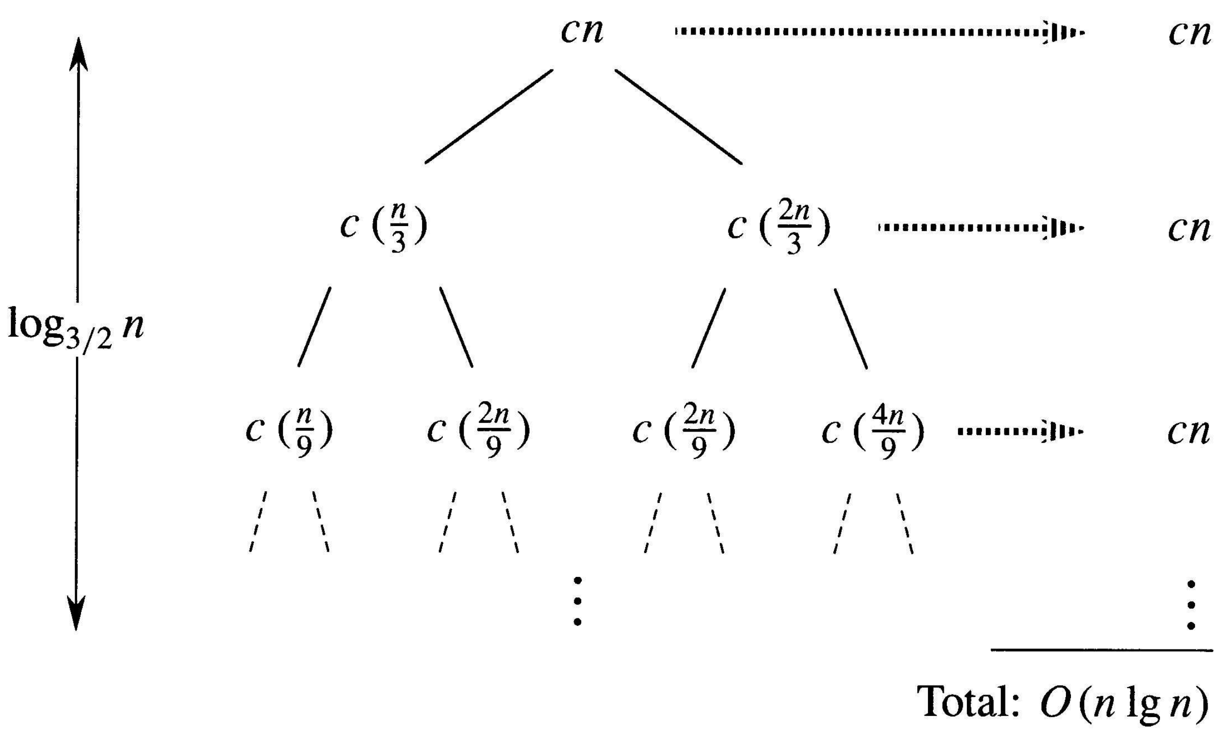
$$T(n) = \Theta(n^2)$$

Recursion Tree 2

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

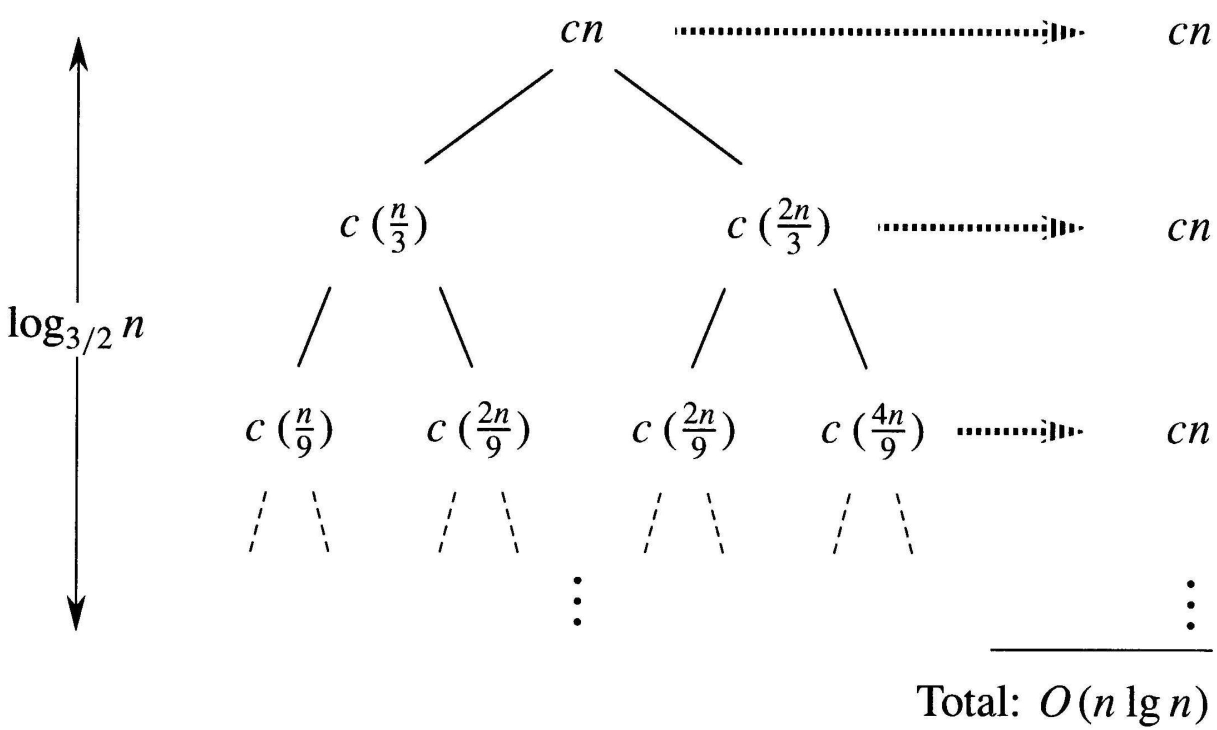


Recursion Tree 2



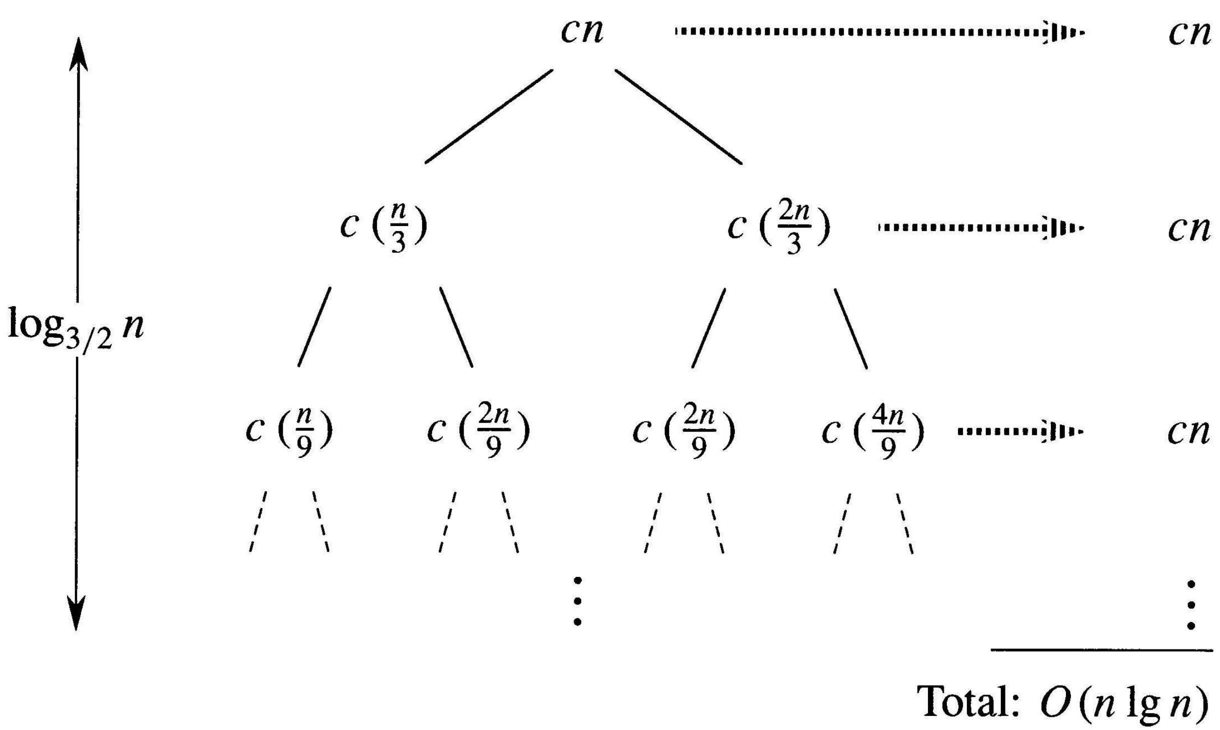
Why is the depth of this tree $\log_{\frac{3}{2}} n$?

Recursion Tree 2



Do all paths from the root to tree leaves have the same length?

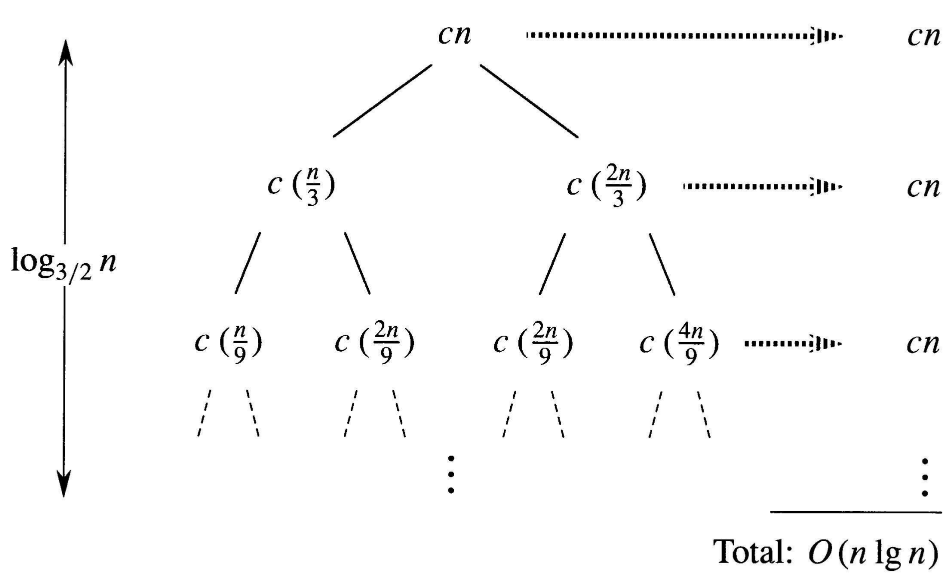
Recursion Tree 2



If all paths were equal to the longest path, what would the cost of the last level be?

Exercise 5.3

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



Prove that $T(n) = O(n \lg n)$ by induction.