Some linear recurrences we talked about:
(1) Towers of Hanoi: $T(n) = 2T(n-1)+1$
(2) Stair climbing: $S(n) = S(n-1) + S(n-2)$
(3) Pandemic outfits: $C(n) = 2C(n-1) + 2C(n-2)$

12/5/23

Lec 27 Handout: Complexity & Divide-and-Conquer algorithms -- ANSWERS

Runtime Recap

$O(1)$, $\log n$, n, $n \log n$, n^2, n^3, (maybe n^4, …), 2^n, $n!$

Grows slowly (better) — Grows quickly (worse)

Consider positive-valued functions $f_1(n) = \Theta(g_1(n))$ and $f_2(n) = \Theta(g_2(n))$.

- **Addition**

$$(f_1 + f_2)(n) = \Theta(\max(g_1(n), g_2(n)))$$

- **Scalar multiplication**

$$af(n) = \Theta(f(n))$$

- **Product**

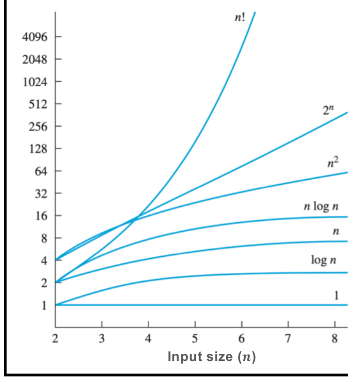
$$(f_1 \cdot f_2)(n) = \Theta(g_1(n) \cdot g_2(n))$$

Lec 27 Handout: Complexity & Divide-and-Conquer algorithms

Runtime Recap

better — worse

$\Theta(1)$, $\log n$, n, $n \log n$, n^2, n^3, (maybe n^4, …), 2^n, $n!$



Scalar multiplication: *ignore* scalar coefficients
$f(n) = 1000n^3$
is $\Theta(\ n^3\ )$

Addition: keep *largest* term
$f(n) = n^2 + \log n$
is $\Theta(\ n^2\ )$

Product: keep *all* terms
$f(n) = n^2 \log n$
is $\Theta(\ n^2 \log n\ )$

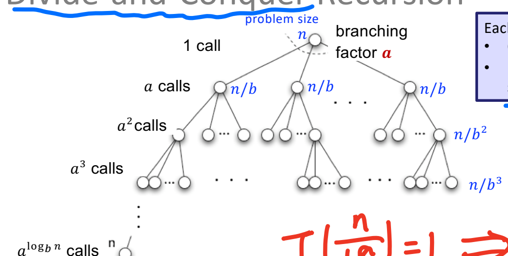
Divide-and-Conquer Recursion

$$\boldsymbol{T(n) = aT(n/b) + \Theta(n^d)}$$

- Divide-and-conquer breaks big problem $\boldsymbol{T(n)}$
  - into $\boldsymbol{a}$ smaller problems,
  - each one of size $\boldsymbol{n/b}$,
  - with cost $\boldsymbol{\Theta(n^d)}$ to put the smaller results together.
- Keep breaking smaller problems into even smaller ones
  - Until the size of each subproblem is **1**
    - until $\boldsymbol{T\left(\frac{n}{b^k}\right) = T(1)}$
  - Depth of tree = $\boldsymbol{k = \log_b n}$

分治

Divide-and-Conquer Recursion



Each recursive call
- Creates a subproblems
- Reduces size of individual subproblems by factor of b

$T\left(\frac{n}{b^a}\right) = 1 \Rightarrow a = \log_b n$

$\boldsymbol{T(n) = aT(n/b) + \Theta(n^d)}$

Key features:
- Number of subproblems: a
- Size of subproblems: n/b
- Work needed to combine results: $\Theta(n^d)$

Note:
Rosen uses a slightly different version of the Master Theorem, with
- $T(n) = aT(n/b) + \boldsymbol{cn^d}$
- Big-O instead of Big-Θ for the 3 cases of runtimes for $T(n)$
  - Ex: $T(n) = \boldsymbol{O}(n^d)$ if $(a/b^d) < 1$

Proof : next page

Master Theorem Practice

The Master Theorem: If $T(n) = aT(n/b) + \Theta(n^d)$, then

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } (a/b^d) < 1 \\ \Theta(n^d \log n) & \text{if } (a/b^d) = 1 \\ \Theta(n^{\log_b a}) & \text{if } (a/b^d) > 1 \end{cases}$$

Recurrence	Big-Θ of F(n)
1. $F(n) = 2F(n/4) + \sqrt{n}$	$\Theta(\sqrt{n}\log(n))$
2. $F(n) = 8F(n/2) + 5n^2$	$\Theta(n^3)$
3. $F(n) = 4F(n/2) + 2n^3$	$\Theta(n^3)$
4. $F(n) = 2F(n/2) + 2$	$\Theta(n)$

1. $a:2,\ b:4,\ d:\frac{1}{2}$ $\quad \frac{a}{b^d} = 1 \Rightarrow n^d \log n \Rightarrow \sqrt{n}\log n$
2. $a:8,\ b:2,\ d:2 \Rightarrow \frac{a}{b^d} = 2 \Rightarrow n^{\log_2 8} = n^3$
3. $a:4,\ b:2,\ d:3 \Rightarrow \frac{a}{b^d} = \frac{1}{2} \Rightarrow n^d = n^3$
4. $a:2,\ b:2,\ d:0 \Rightarrow \frac{a}{b^d} = 2 \Rightarrow n^{\log_2 2} = n$

Master Theorem with Pseudocode

Find the Big-Θ runtime of the following algorithm:

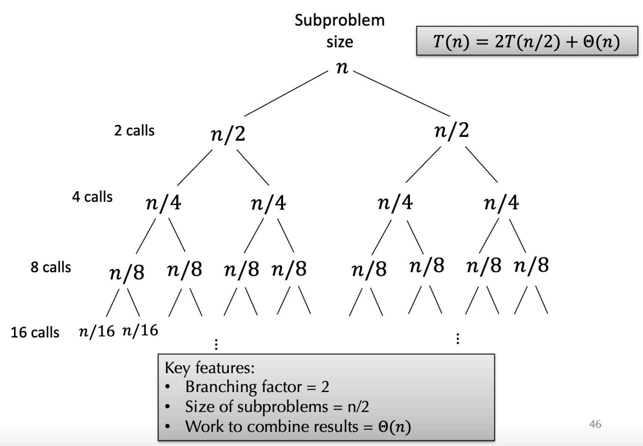
```
procedure farewell(n: integer)
    if n=0, stop
    farewell(n/3)
    farewell(n/3)
    farewell(n/3)
    farewell(n/3)

    for i := 1 to n/4
        for j := 1 to n
            print "good luck with finals!"
```

4 subcalls of size n/3

n ; $n/4 \equiv n$ $\Rightarrow \Theta(n^2)$

$T(n) = 4T(n/3) + \Theta(n^2)$
- Master Theorem with: $a = 4, b = 3, d = 2$
- $(a/b^d) = 4/3^2 = 4/9 < 1$
- $T(n) = \Theta(n^d) = \Theta(n^2)$

Merge Sort Recursion



- The recurrence for MergeSort:
  - $T(n) = 2T(n/2) + \Theta(n)$ and $T(1) = 1$.
  - $T(n) = aT(n/b) + \Theta(n^d)$
- So $a = 2,\ b = 2,\ d = 1$. Therefore $a/b^d = 1$.
- The Master Theorem tells us that
  - $T(n)$ is in $\Theta(n^d \log n) = \Theta(n \log n)$.

Find the Big-Θ runtime of the following algorithm:

```
procedure farewell(n: integer)
    if n=0, stop
    farewell(n/3)
    farewell(n/3)
    farewell(n/3)
    farewell(n/3)

    for i := 1 to n/4
        for j := 1 to n
            print "good luck with finals!"
```

4 subcalls of size n/3

n ; $n/4 \equiv n$ $\Rightarrow \Theta(n^2)$

$T(n) = 4T(n/3) + \Theta(n^2)$
- Master Theorem with: $a = 4, b = 3, d = 2$
- $(a/b^d) = 4/3^2 = 4/9 < 1$
- $T(n) = \Theta(n^d) = \Theta(n^2)$

- $T(n) = aT(n/b) + n^d$ (recursive form)

$$T(n) = n^d + a\left(\frac{n}{b}\right)^d + a^2\left(\frac{n}{b^2}\right)^d + \cdots + a^{\log_b n}(1)^d$$

$$= n^d + a\left(\frac{n}{b}\right)^d + a^2\left(\frac{n}{b^2}\right)^d + \cdots + a^{\log_b n}\left(\frac{n}{b^{\log_b n}}\right)^d$$

$$= n^d \underbrace{\left(1 + \frac{a}{b^d} + \left(\frac{a}{b^d}\right)^2 + \cdots + \left(\frac{a}{b^d}\right)^{\log_b n}\right)}_{\text{geometric sum with base } (a/b^d)}$$

1° $\frac{a}{b^d} < 1$

$\Rightarrow$ geometric series

$\Rightarrow \sum_{i=0}^{\log_b n} \left(\frac{a}{b^d}\right)^i$

$\approx \frac{1}{1-\frac{a}{b^d}}$ (const)

$\Rightarrow T(n) = cn^d = \Theta(n^d)$,

- If $\boldsymbol{a/b^d} < \mathbf{1}$
  - the terms in the sum are getting smaller, so the **first** term dominates the sum
  - $\Theta(T(n)) = \Theta(n^d(1)) = \Theta(n^d)$

- $T(n) = aT(n/b) + n^d$ (recursive form)

$$T(n) = n^d + a\left(\frac{n}{b}\right)^d + a^2\left(\frac{n}{b^2}\right)^d + \cdots + a^{\log_b n}(1)^d$$

$$= n^d + a\left(\frac{n}{b}\right)^d + a^2\left(\frac{n}{b^2}\right)^d + \cdots + a^{\log_b n}\left(\frac{n}{b^{\log_b n}}\right)^d$$

$$= n^d \underbrace{\left(1 + \frac{a}{b^d} + \left(\frac{a}{b^d}\right)^2 + \cdots + \left(\frac{a}{b^d}\right)^{\log_b n}\right)}_{\text{geometric sum with base } (a/b^d)}$$

2° $\frac{a}{b^d} = 1$

$\sum_{i=0}^{\log_b n} \left(\frac{a}{b^d}\right)^i = \log_b n$

$\Rightarrow T(n) = n^d \log_b n$

- If $\boldsymbol{a/b^d} = \mathbf{1}$
  - the terms in the sum are all 1, so the **number of terms** impacts $\Theta(T(n))$
  - $\Theta(T(n)) = \Theta(n^d(1 + 1 + 1 + \cdots + 1)) = \Theta(n^d \log_b n)$

- $T(n) = aT(n/b) + n^d$ (recursive form)

$$T(n) = n^d + a\left(\frac{n}{b}\right)^d + a^2\left(\frac{n}{b^2}\right)^d + \cdots + a^{\log_b n}(1)^d$$

$$= n^d + a\left(\frac{n}{b}\right)^d + a^2\left(\frac{n}{b^2}\right)^d + \cdots + a^{\log_b n}\left(\frac{n}{b^{\log_b n}}\right)^d$$

$$= n^d \underbrace{\left(1 + \frac{a}{b^d} + \left(\frac{a}{b^d}\right)^2 + \cdots + \left(\frac{a}{b^d}\right)^{\log_b n}\right)}_{\text{geometric sum with base } (a/b^d)}$$

3° $\frac{a}{b^d} > 1$

$S = \Theta\left(\left(\frac{a}{b^d}\right)^{\log_b n}\right) = O\left(\left(\frac{a}{b^d}\right)^{\log_b n}\right)$

(last term dominates)

$\Rightarrow T(n) = n^d S$

$= \Theta\left(n^d \left(\frac{a}{b^d}\right)^{\log_b n}\right)$

$= \Theta\left(n^d \frac{a^{\log_b n}}{n^d}\right) = \Theta\left(a^{\log_b n}\right) = \underline{\Theta\left(a^{\frac{\log_a n}{\log_a b}}\right) = \Theta\left(n^{\frac{1}{\log_a b}}\right) = \Theta\left(n^{\log_b a}\right)}$

- If $\boldsymbol{a/b^d} > \mathbf{1}$
  - the terms in the sum are getting larger, so the **last** term dominates the sum
  - $\Theta(T(n)) = \Theta\left(n^d \left(\frac{a}{b^d}\right)^{\log_b n}\right) = \ldots = \Theta(n^{\log_b a})$

☆ $a^{\log_b n} = n^{\log_b a}$

Closed Form for $T(n)$

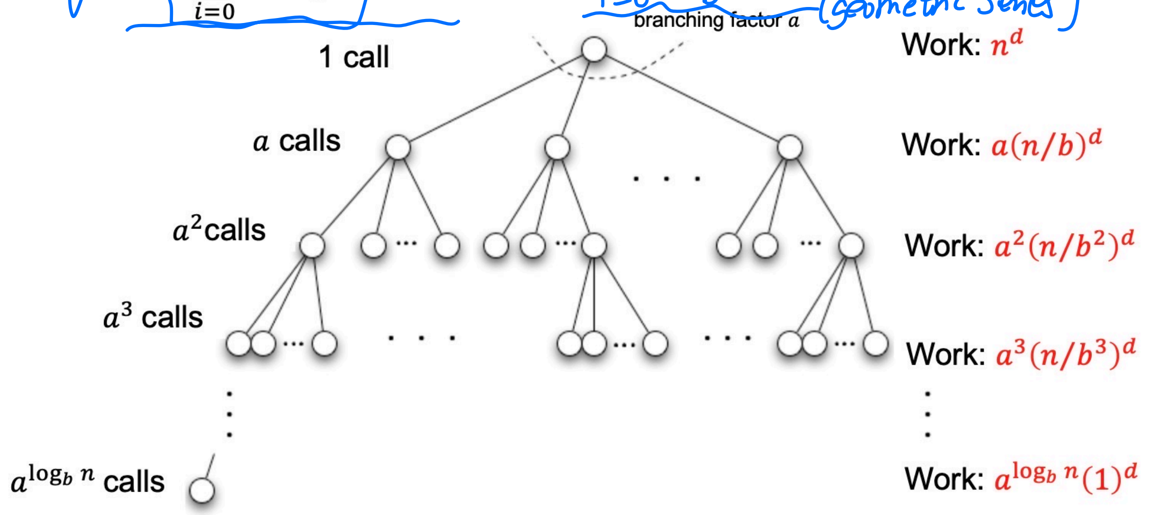
- $T(n) = aT(n/b) + n^d$ (recursive form)

$$T(n) = n^d + a\left(\frac{n}{b}\right)^d + a^2\left(\frac{n}{b^2}\right)^d + \cdots + a^{\log_b n}(1)^d \quad \text{(closed form)}$$

$$\star \quad = \boxed{\sum_{i=0}^{\log_b n} a^i \left(\frac{n}{b^i}\right)^d} = n^d \cdot \sum_{i=0}^{\log_b n} \left(\frac{a}{b^d}\right)^i$$

(geometric series)

Total work at this level



Analyzing complexity: Example 2

- $T(n) = 5T(n/4) + n.$ Find O(T(n))

Depth	# of calls	Size of calls	Work/call	Total work/depth
0	1	n	n	n
k	5^k	$n/4^k$	$n/4^k$	$n(5/4)^k$
$x = \log_4 n$	$5^{\log_4 n}$	$1 = \frac{n}{4^x}$	1	$n(5/4)^{\log_4 n} = 5^{\log_4 n}$

$$T(n) = n\left[1 + \tfrac{5}{4} + \left(\tfrac{5}{4}\right)^2 + \left(\tfrac{5}{4}\right)^3 + \cdots + \left(\tfrac{5}{4}\right)^x\right] \quad \text{(Geom. Sum, } r = 5/4\text{)}$$

$$= n\left[\left(\tfrac{5}{4}\right)^x + \left(\tfrac{5}{4}\right)^{x-1} + \cdots + \left(\tfrac{5}{4}\right)^3 + \left(\tfrac{5}{4}\right)^2 + \tfrac{5}{4} + 1\right]$$

Trick: reverse the order of the terms

$$< n\left(\tfrac{5}{4}\right)^x\left[1 + \tfrac{4}{5} + \left(\tfrac{4}{5}\right)^2 + \left(\tfrac{4}{5}\right)^3 + \cdots + \left(\tfrac{4}{5}\right)^{9999} + \cdots\right] \quad \text{(Geom. Sum, } r = 4/5\text{)}$$

$$= n\left(\tfrac{5}{4}\right)^x \frac{1}{1-4/5}$$

$$= 5n\left(\tfrac{5}{4}\right)^x$$

$$= 5 \cdot 5^{\log_4 n} \quad (x = \log_4 n)$$

$$= 5 \cdot n^{\log_4 5} \quad (z^{\log_b y} = y^{\log_b z})$$

$T(n)$ is $O(n^{\log_4 5})$

Also $\Theta(n^{\log_4 5})$