Design and Analysis of Algorithms I

Master Method

Proof (Part II)

The Story So Far/Case 1

Total work: $\le cn^d \times \underbrace{\sum_{j=0}^{\log_b n} \left(\frac{a}{b^d}\right)^j}_{=(\log_b n + 1)}$ (*)

$= 1$ for all j

$= 1$

If $a = b^d$, then

$$(*) = cn^d(\log_b n + 1)$$

$$= O(n^d \log n)$$

(end Case 1)

Basic Sums Fact

For $r \neq 1$, we have

$$1 + r + r^2 + r^3 + \cdots + r^k = \frac{r^{k+1} - 1}{r - 1}$$

Proof: by induction (you check).

Upshot:

(1) if $r < 1$ is constant, RHS is $\leq \frac{1}{1-r}$ = a constant ← independent of k

(i.e., 1st term of sum dominates)

(2) if $r > 1$ is constant, RHS is $\leq r^k \cdot \left(1 + \frac{1}{r-1}\right)$ ← independent of k

(i.e., last term of sum dominates)

Case 2

$$\text{Total work:} \quad \leq cn^d \times \sum_{j=0}^{\log_b n} \left(\frac{a}{b^d}\right)^j \qquad (*)$$

$\frac{a}{b^d} := r$

$\leq$ a constant (independent of n) [by basic sums fact]

If $a < b^d$ [RSP < RWS]

$= O(n^d)$

[total work dominated by top level]

Tim Roughgarden

Case 3

Total work: $\leq cn^d \times \sum_{j=0}^{\log_b n} \left(\frac{a}{b^d}\right)^j$ (*)

$\frac{a}{b^d} := r > 1$

$\hookrightarrow \leq$ constant * largest term

If $a > b^d$ [RSP > RWS]

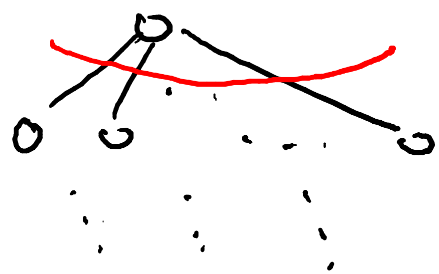
then (*) $= O\left(n^d \cdot \left(\frac{a}{b^d}\right)^{\log_b n}\right)$

<u>Note</u>: $b^{-d\log_b n} = \left(\cancel{b}^{\cancel{\log_b} n}\right)^{-d} = n^{-d}$

<u>So</u>: (*) $= O\left(a^{\log_b n}\right)$

Tim Roughgarden

level 0



a children

1

level $\log_b n$

\# leaves $= a^{\log_b n}$

Which of the following quantities is equal to $a^{\log_b n}$?

- The number of levels of the recursion tree.
- The number of nodes of the recursion tree.
- The number of edges of the recursion tree.
- The number of leaves of the recursion tree.

Case 3 continued

Total work: $\leq cn^d \times \sum_{j=0}^{\log_b n} (\frac{a}{b^d})^j$ (*)

So: (*) $= O(a^{\log_b n}) = O(\#\text{leaves})$

Note: $a^{\log_b n} = n^{\log_b a}$

more intuitive

simpler to apply

[since $(\log_b n)(\log_b a) = (\log_b a)(\log_b n)$]

[end Case 3]

QED!

The Master Method

If $T(n) \leq aT\left(\frac{n}{b}\right) + O(n^d)$

then

$$T(n) = \begin{cases} O\left(n^d \log n\right) & \text{if } a = b^d \quad \text{(Case 1)} \\ O(n^d) & \text{if } a < b^d \quad \text{(Case 2)} \\ O(n^{\log_b a}) & \text{if } a > b^d \quad \text{(Case 3)} \end{cases}$$