Part4-The Master Method

Completed on	
Key videos	
Note	

Note

▼ Motivation

- 重新分析整数乘法

Integer Multiplication Revisited

Motivation : potentially useful algorithmic ideas often need mathematical analysis to evaluate

Recall : grade-school multiplication algorithm uses $\theta(n^2)$ operation to multiply two n-digit numbers

- 一般算法分析：分为四个子问题，加减都是O(n)复杂度！

A Recursive Algorithm

Recursive approach

Write $x = 10^{n/2}a + b \qquad y = 10^{n/2}c + d$

[where a,b,c,d are n/2 – digit numbers]

So :

$$x \cdot y = 10^n ac + 10^{n/2}(ad + bc) + bd \qquad (*)$$

Algorithm#1 : recursively compute ac,ad,bc,bd, then compute (*) in the obvious way.

T(n) = maximum number of operations this algorithm needs to multiply two n-digit numbers

Recurrence : express T(n) in terms of running time of recursive calls.

Base Case : T(1) <= a constant.

For all n > 1 : $T(n) \le 4T(n/2) + O(n)$

Work done here

Work done by recursive calls

- 优化大数乘法：三次乘法，加上一下额外的加减O(n)复杂度

A Better Recursive Algorithm

Algorithm #2 (Gauss) : recursively compute ac[(1)], bd[(2)], (a+b)(c+d)[(3)] [recall ad+bc = (3) – (1) – (2)]

New Recurrence :

Base Case : T(1) <= a constant

For all n>1 : $T(n) \le 3T(n/2) + O(n)$

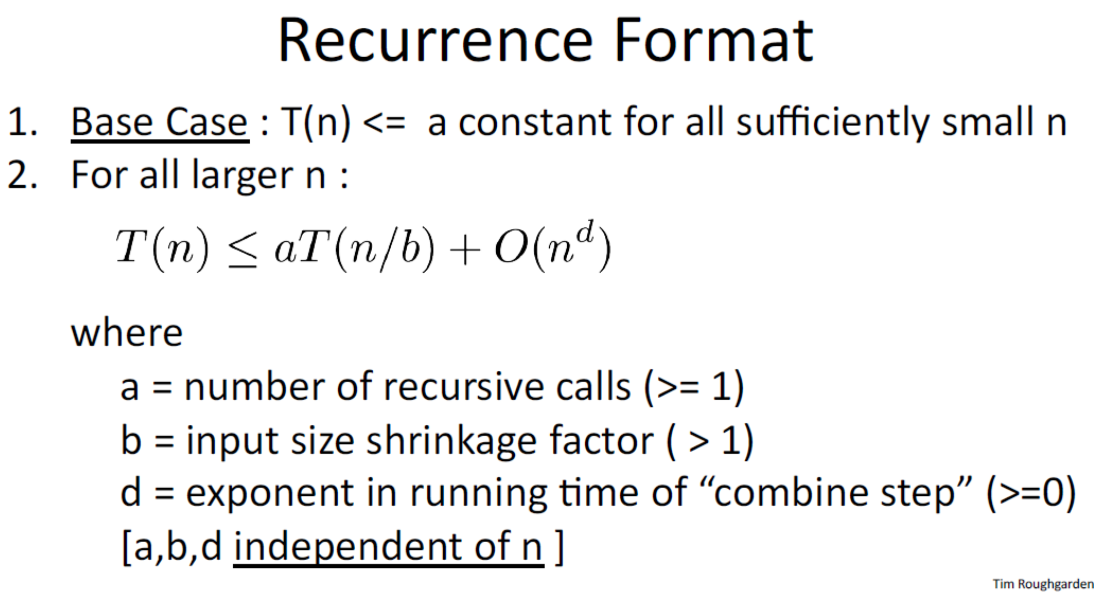
Work done here

Work done by recursive calls

▼ Formal Statement

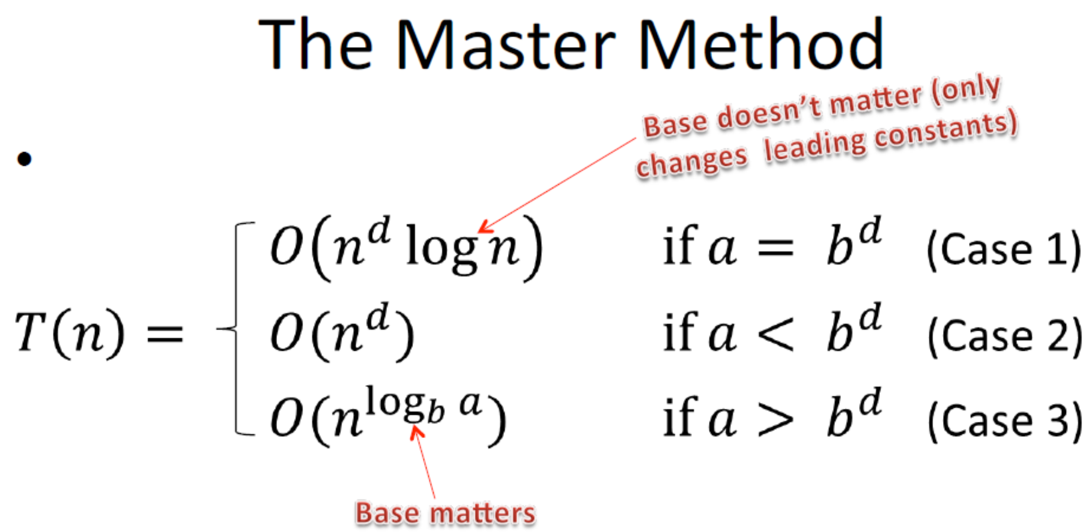
- 基本分析原理：假设所有子问题尺寸等价(all subproblems have equal size)
- 基本形式：

  除了跟输入大小n以外，有三个关键参数：递归子问题个数a；子问题缩减因子b；以及合并阶段指数d



- The Master Method

  三种情况的概括性分析



▼ Examples

▼ Proof I

▼ Interpretation of the 3 Cases

▼ Proof II

References

- Master Method

[主定理](https://zh.wikipedia.org/wiki/%E4%B8%BB%E5%AE%9A%E7%90%86)
https://zh.wikipedia.org/wiki/%E4%B8%BB%E5%AE%9A%E7%90%86

[Master theorem (analysis of algorithms)](https://en.wikipedia.org/wiki/Master_theorem_(analysis_of_algorithms))
https://en.wikipedia.org/wiki/Master_theorem_(analysis_of_algorithms)

常用算法中的应用 [编辑]

算法	递回关系式	运算时间	备注
二分搜寻算法	$T(n) = T\left(\frac{n}{2}\right) + \Theta(1)$	$\Theta(\log n)$	情形二（$k = 0$）
二叉树遍历	$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(1)$	$\Theta(n)$	情形一
最佳排序矩阵搜索(已排好序的二维矩阵)	$T(n) = 2T\left(\frac{n}{2}\right) + O(\log n)$	$\Theta(n)$	
合并排序	$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$	$\Theta(n \log n)$	情形二（$k = 0$）

- binary search

[二分搜尋演算法](https://zh.wikipedia.org/wiki/%E4%BA%8C%E5%88%86%E6%90%9C%E5%B0%8B%E6%BC%94%E7%AE%97%E6%B3%95)
https://zh.wikipedia.org/wiki/%E4%BA%8C%E5%88%86%E6%90%9C%E5%B0%8B%E6%BC%94%E7%AE%97%E6%B3%95

[Binary search algorithm](https://en.wikipedia.org/wiki/Binary_search_algorithm)
https://en.wikipedia.org/wiki/Binary_search_algorithm