

数据结构

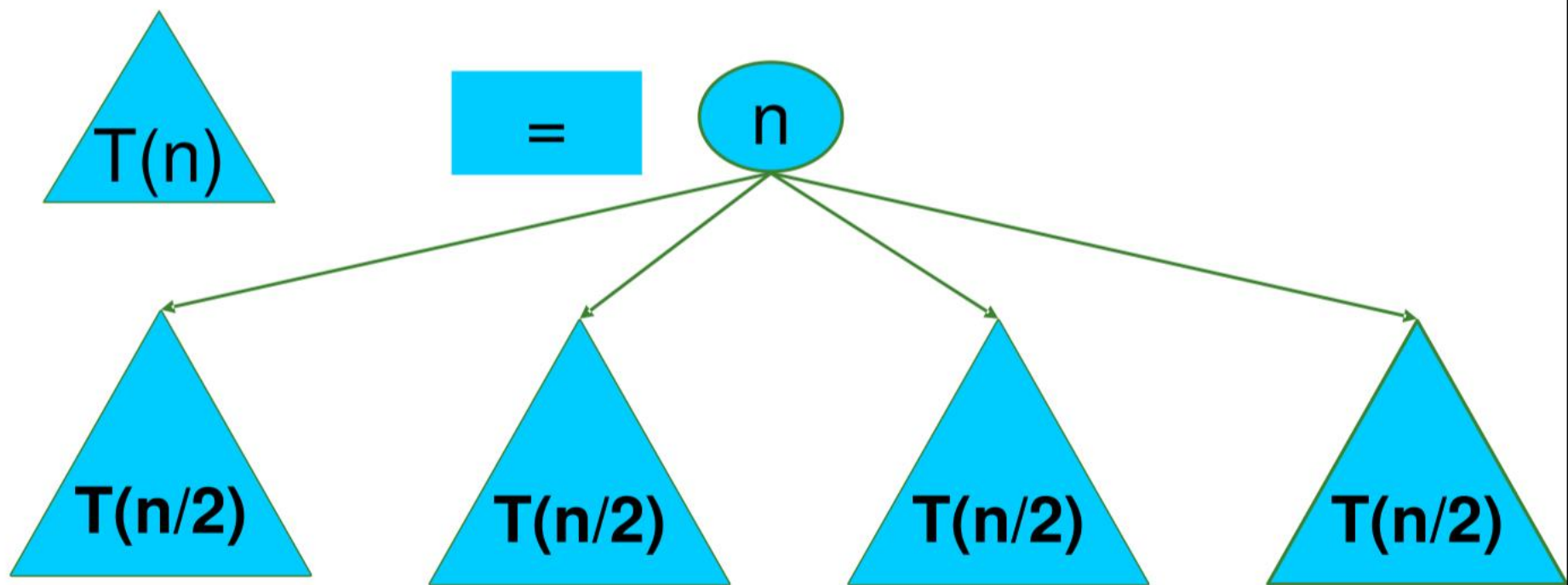
深圳技术大学
大数据与互联网学院

What is dynamic programming

- 与分治法类似, 动态规划也是通过组合子问题的解来求解问题.
- 分治算法将问题划分成独立子问题, 递归地解决这些子问题, 然后组合这些子问题的解来求解原始问题.
- **If these subproblems are not independent, what will happen?**

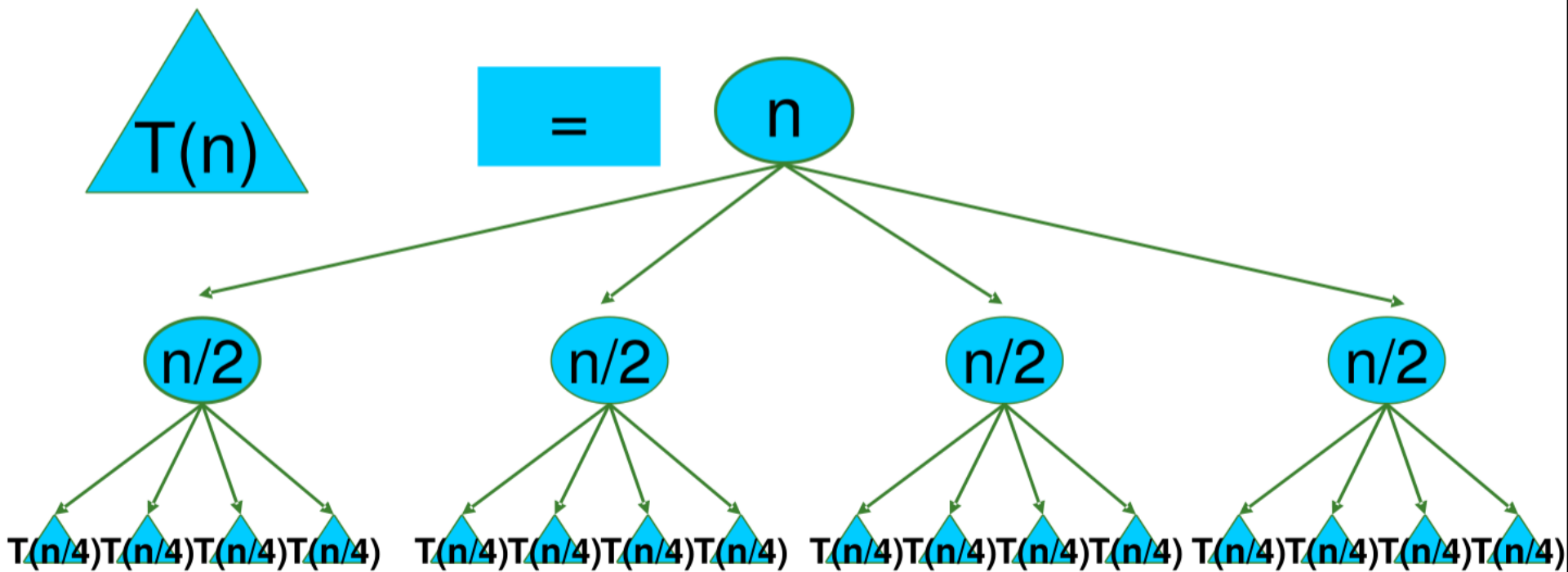
算法总体思想

- 动态规划算法与分治法类似，其基本思想也是将待求解问题分解成若干个子问题



算法总体思想

- 但是经分解得到的子问题往往不是互相独立的。不同子问题的数目常常只有多项式量级。在用分治法求解时，有些子问题被重复计算了许多次。



算法总体思想

- 如果能够保存已解决的子问题的答案，而在需要时再找出已求得的答案，就可以避免大量重复计算，从而得到多项式时间算法。

**Those who cannot remember the past
are doomed to repeat it.**

（无法记取教训者必重蹈覆辙）

-----George Santayana,
The life of Reason,
Book I: Introduction and
Reason in Common

Sense (1905)

What is dynamic programming

什么是动态规划？

- 当子问题发生重叠时, 分治法做了很多不必要的工作——重复对重叠的子问题进行求解.
- 动态规划算法对每个子问题求解一次, 然后将结果保存在一张表里面, 这样可以避免每个已求解子问题的重复计算.
- 对于Fibonacci序列, 一个明显的方法是从 $f(1)$ 开始自底向上地计算到 $f(n)$, 只需要 $\Theta(n)$ 时间和 $\Theta(1)$ 空间.
- 和前面的方法相比, 可以很大程度降低时间复杂度.

Fibonacci sequence(序列)

- **Fibonacci**序列定义如下:
 - 1. procedure $f(n)$
 - 2. if $n=1$ or $n=2$ then return 1
 - 3. else return $f(n-1)+f(n-2)$
- 这种递归形式有简洁、容易书写和容易查错等优点,最主要是它的抽象性.
- 但是它远不是有效的算法.
 - 算法复杂性: $\Theta(\phi^n)$
 - Why???

Matrix chain multiplication 矩阵链相乘

- 设有4个矩阵A, B, C, D, 它们的维数分别是
A:50×10 B:10×40 C:40×30 D:30×5, 共有5种加括号的方式:
- (A((BC)D))
 - 乘法次数: 16000
- (A(B(CD)))
 - 乘法次数: 10500
- ((AB)(CD))
 - 乘法次数: 36000
- (((AB)C)D)
 - 乘法次数: 87500
- ((A(BC))D)
 - 乘法次数: 34500

$M_1:5 \times 10$ $M_2:10 \times 4$ $M_3:4 \times 6$ $M_4:6 \times 10$
 $M_5:10 \times 2$

$C[1,1]=0$	$C[1,2]=200$	$C[1,3]=320$	$C[1,4]=620$	$C[1,5]=348$
	$C[2,2]=0$	$C[2,3]=240$	$C[2,4]=640$	$C[2,5]=248$
		$C[3,3]=0$	$C[3,4]=240$	$C[3,5]=168$
			$C[4,4]=0$	$C[4,5]=120$
				$C[5,5]=0$

图7.3 矩阵链乘算法的一个例子