

问题求解-论题1-5

-数据与数据结构

陶先平 陈道蓄

2015年10月28日

问题1：为什么每个数据都应该有个“类型”和它对应？

其实，计算机操纵的对象（数据）“类别”在表现形式上可以统一，但是：

It is beneficial, however, to keep such types separate, not only for clarity and good order, but also because each type admits its own special set of allowed operations, or actions.

问题2： 变量是不是量？

- 我们该如何理解以下程序语句？

$X=X+1;$

- 从计算机的视角出发，这条语句“背后”我们能看到什么？
 - 变量是空间的代号
 - 对数据的操纵：取，存（改变）

变量、数据、类型

- 变量和数据
 - 变量是用于跟踪、操纵几乎所有数据的简单(通用)工具
 - 数据是有“内存”地址的，变量名和地址是什么关系？
- 变量和数据类型
 - 类型定义了变量的变化范围
 - 类型定义了计算对变量的操作方式

你能看清这个程序片段吗？

```
int *p1, *p2;
```

```
*p1:=44;
```

```
*p2:=99;
```

```
P1:=P2;
```

```
Print(*p1,*p2);
```

```
int *p1, *p2;
```

```
*p1:=44;
```

```
*p2:=99;
```

```
*p1:=*p2;
```

```
Print(*p1,*p2);
```

指针到底是什么？两个不同的赋值到底区别在哪里？

多个数据的聚集管理而出现的“结构”

- 场景：“一队士兵”
 - 每个士兵有了一个队伍中的唯一“位置”
 - 位置是相对的，位置可以调整（how?）
 - 如果设计一个按照位置进行的“游戏”，给定了位置就指定了人
- 按照上述观点，vector/list/one-dimensional array为什么被称为是一种数据结构，它的“结构性”体现在哪里？

改变“位置”和改变“内容”

- (1) do the following $N - 1$ times:
 - (1.1) $X \leftarrow 1$;
 - (1.2) while $X < N$ do the following:
 - (1.2.1) if $V[X + 1] < V[X]$ then exchange them;
 - (1.2.2) $X \leftarrow X + 1$.

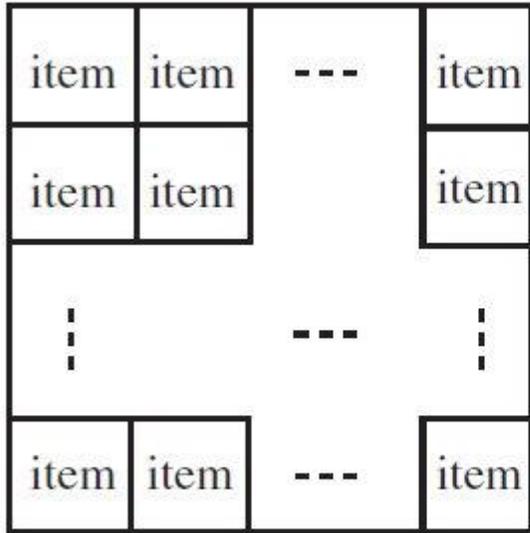
How to exchange them?

如何理解以下文字？

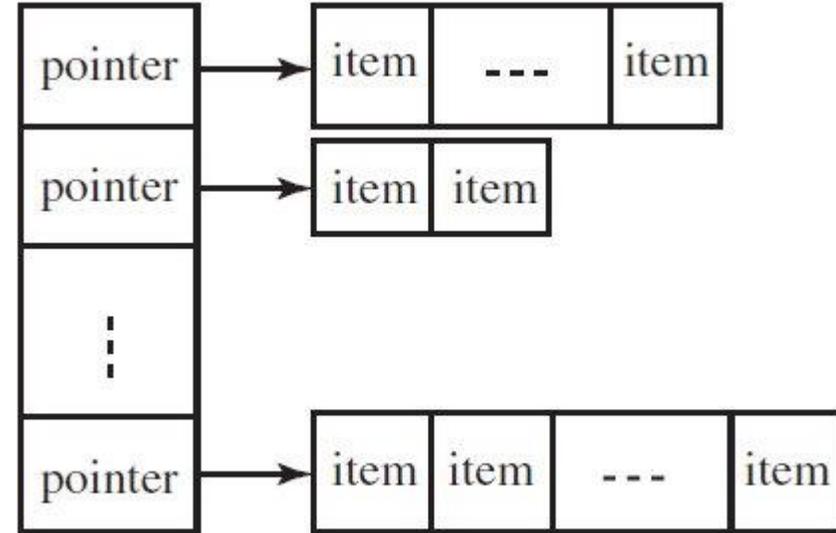
Just as the loop is a control structure for describing lengthy processes, so is a vector a data structure for representing lengthy lists of data items.⁴

数组与循环？

Vector、Vector of vectors和Array有什么区别？

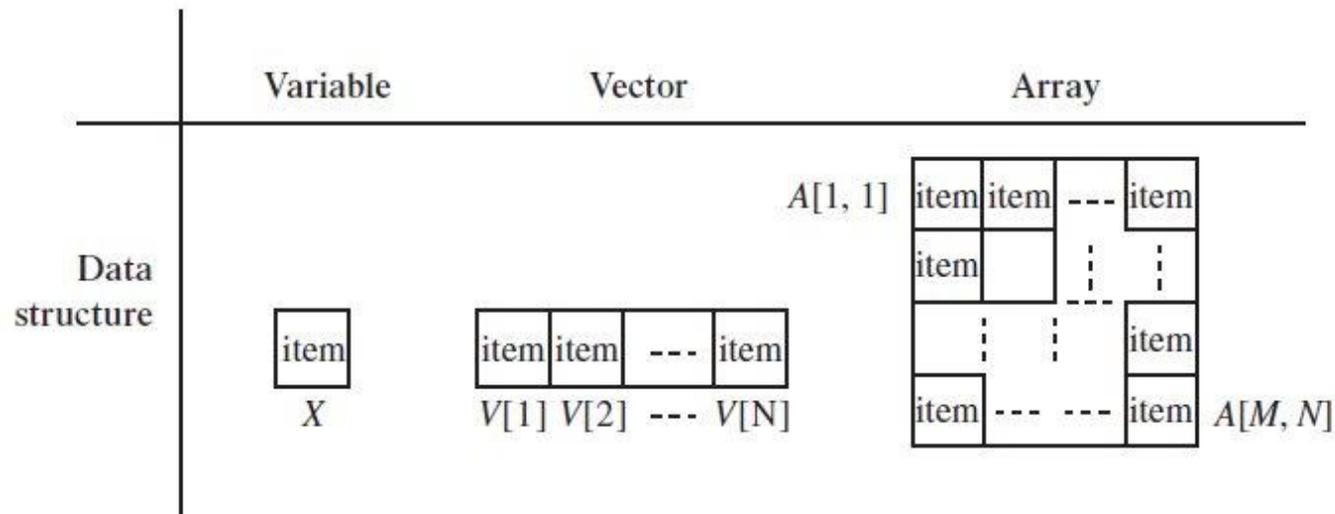


Array



Vector of vectors

问题：如何访问“单个变量”、“向量”和“数组”？



问题：如何遍历“单个变量”、“向量”和“数组”？

问题： 以下两者的区别何在？

- 在书架中取一本书

在一桶纸杯中拿一个杯子



in many applications of vectors and arrays, we do not need the full power provided by indices. Sometimes a list is used just to model a **queue**



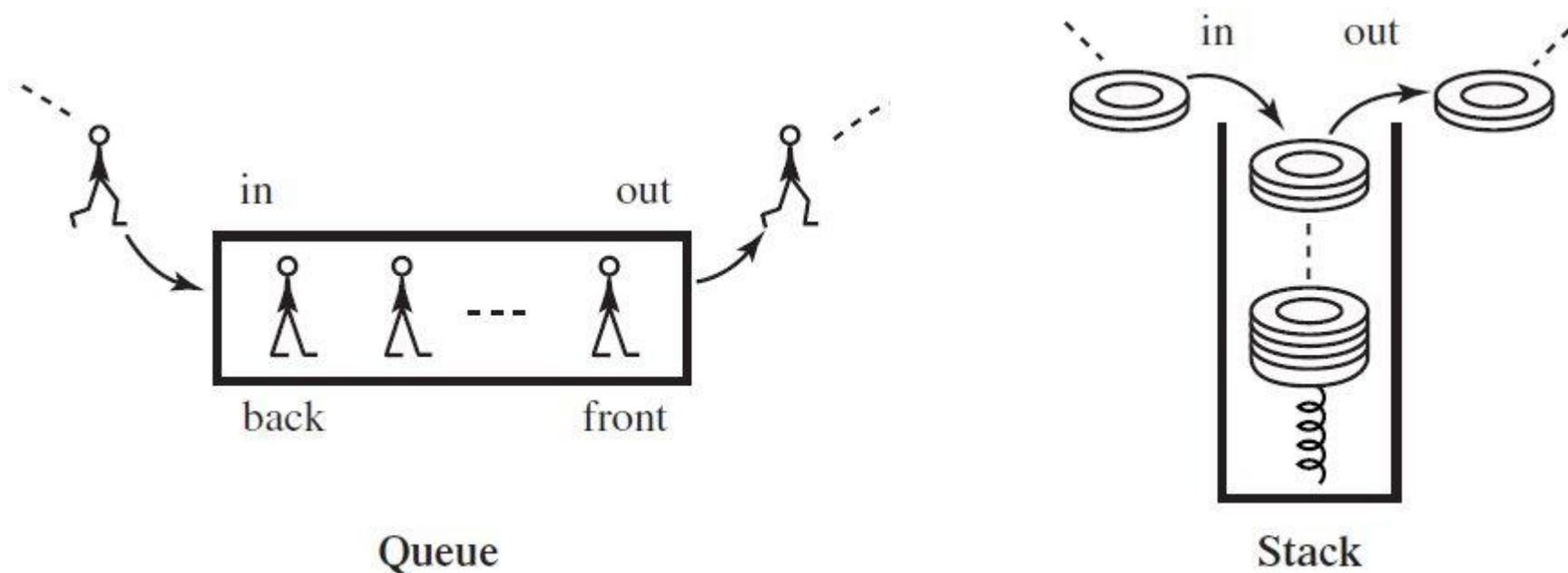
使用时的随意性：任意存取

使用时的受限性：头存尾取

代价更高的数组才能实现如此的随意性

代价较低的队列就能实现受限的存取

两种用途非常广泛的特殊“vector”：队列、栈



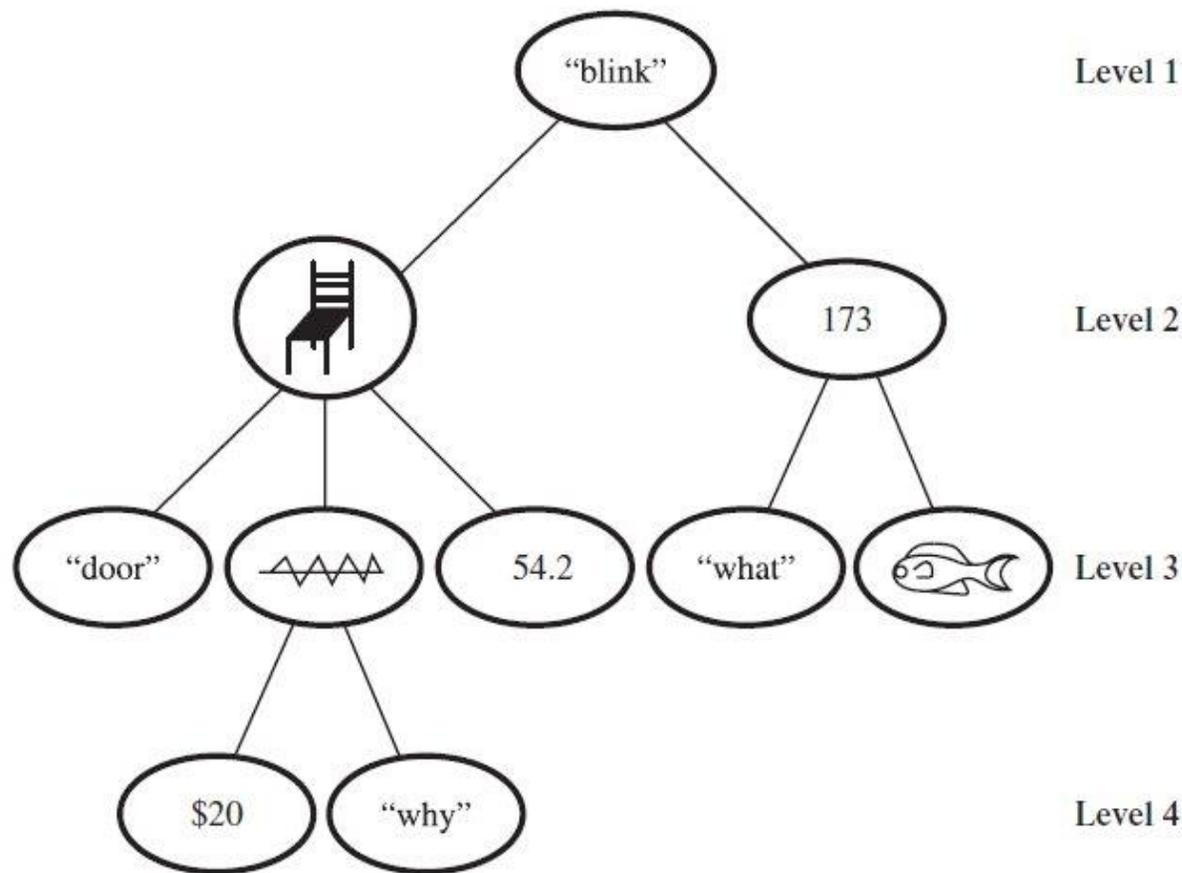
The point in these special cases is that it is worthwhile, at least for reasons of algorithmic clarity, to think of queues and stacks as being in themselves data structures, rather than being merely special kinds of lists. We can then use specially devised elementary instructions such as “add X to queue A ,” or “push X on stack S ,” rather than obscure formulations that explicitly involve indices.

我们再来理解以下文字：

tests associated with them for granted. What we are interested in are the ways algorithms can organize, remember, change, and access collections of data. While control structures serve to tell the processor where it should be going, data structures, and the operations upon them, organize the data items in ways that enable it to do whatever it should do when it gets there.

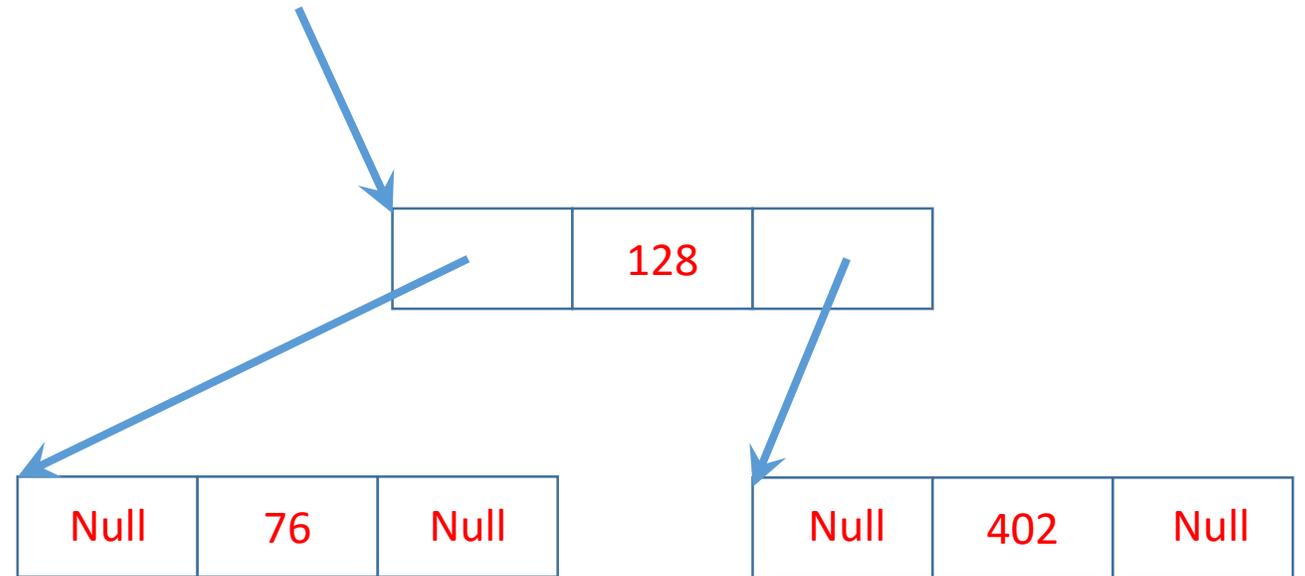
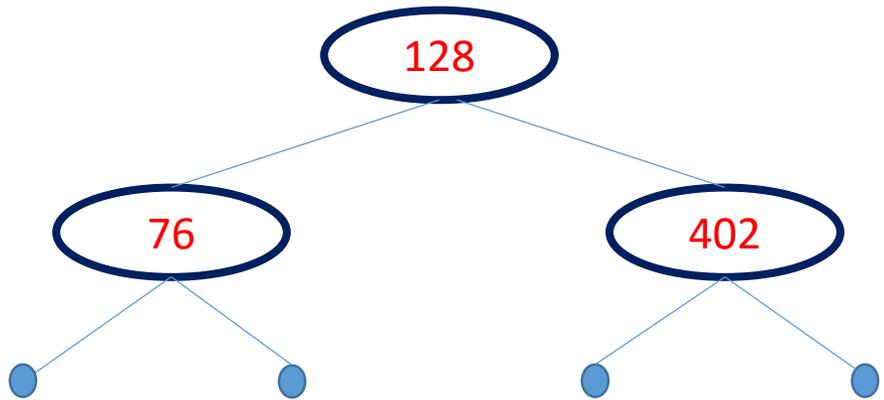
从理解数据结构的“结构”角度出发，哪些词最为关键？

问题：为什么“树”也是一种数据结构？



一种可以用来表达更为复杂的数据间“位置”关系（层次关系）的数据结构

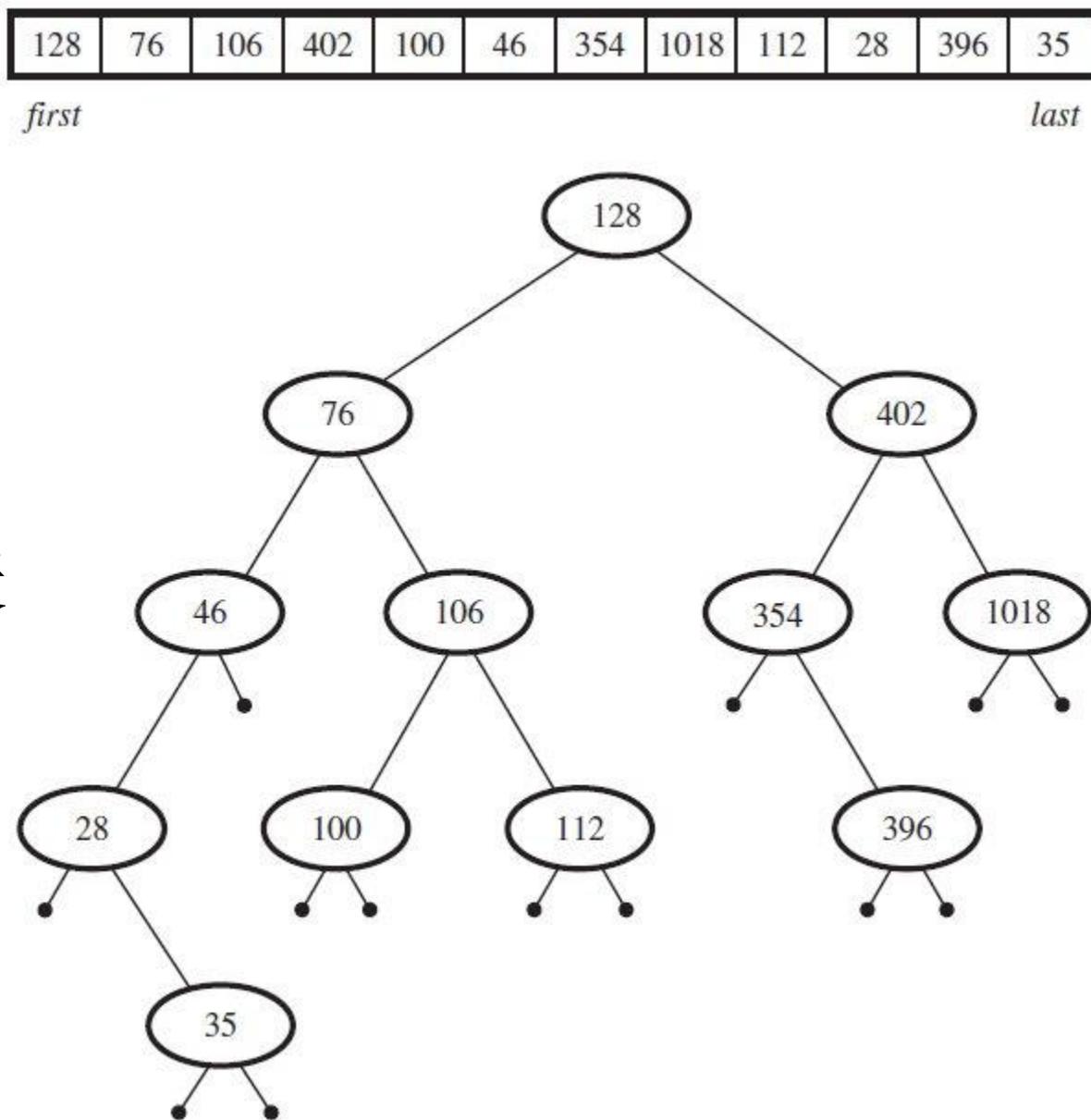
树在内存中是如何实现的？



和在Array中我们可以规定数据的“位置”一样，我们也可以在树中给数据“定位”，进而做一些有趣的事情

树排序：

第一步，将待排序数列表示为“二分搜索树”



相应的算法:

subroutine **second-visit-traversal-of** T :

- (1) if T is empty then return;
- (2) otherwise (i.e., if T is not empty) do the following:
 - (2.1) call **second-visit-traversal-of** $left(T)$;
 - (2.2) output the data element found at the root of T ;
 - (2.3) call **second-visit-traversal-of** $right(T)$;
- (3) return.

你能证明这个算法是正确的吗?

树结构和递归结构有着天然的联系！

什么关系？

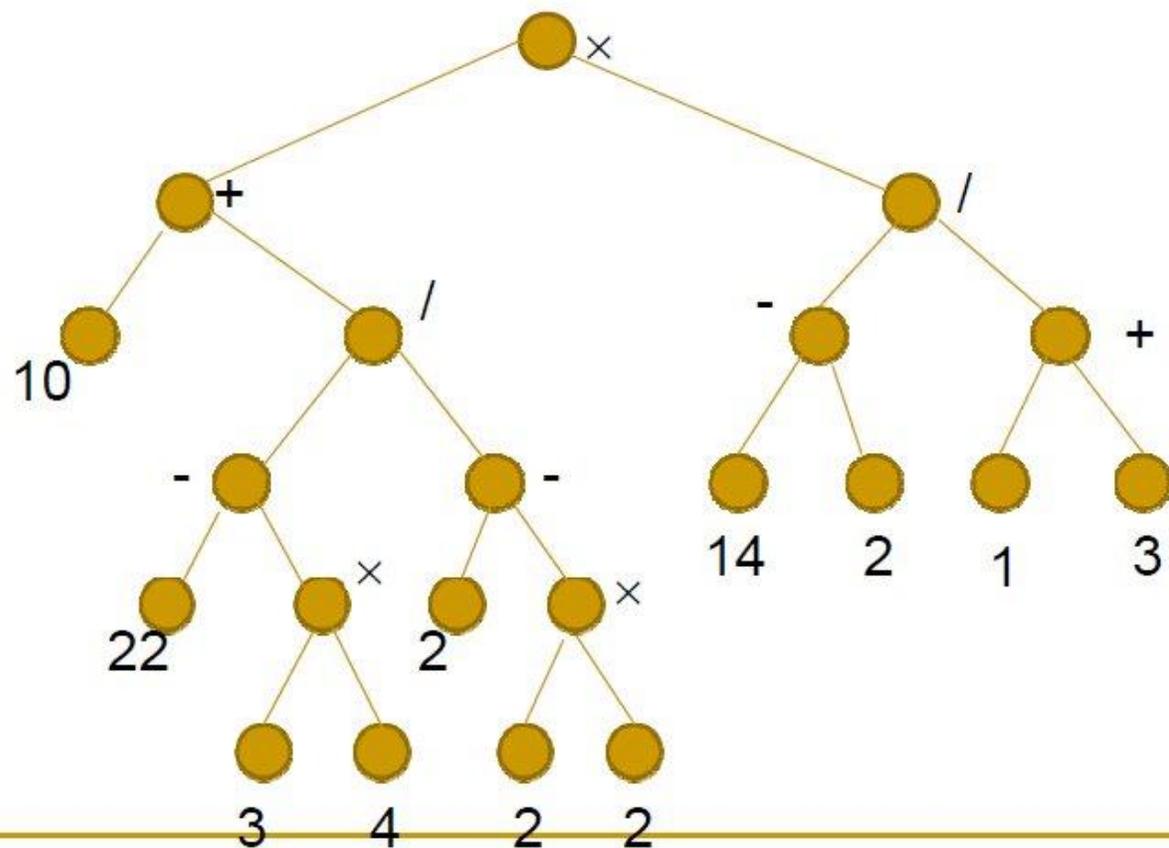
利用树和栈解更复杂的问题

- 如何自动表述一个带括号的算术式的计算序列？
- 例如：

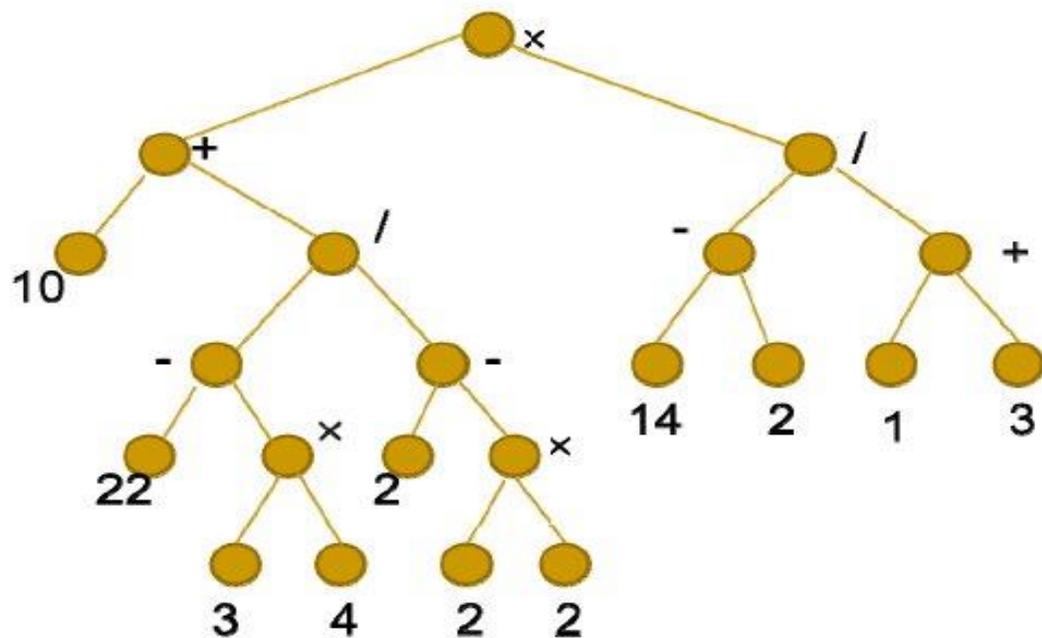
$$(10 + ((22 - 3 \times 4) / 2 - 2 \times 2) \times ((14 - 2) / (1 + 3)))$$

第1步：构造一个表达式树

$$(10 + ((22 - 3 \times 4) / 2 - 2 \times 2) \times ((14 - 2) / (1 + 3)))$$



第2步：遍历这个树



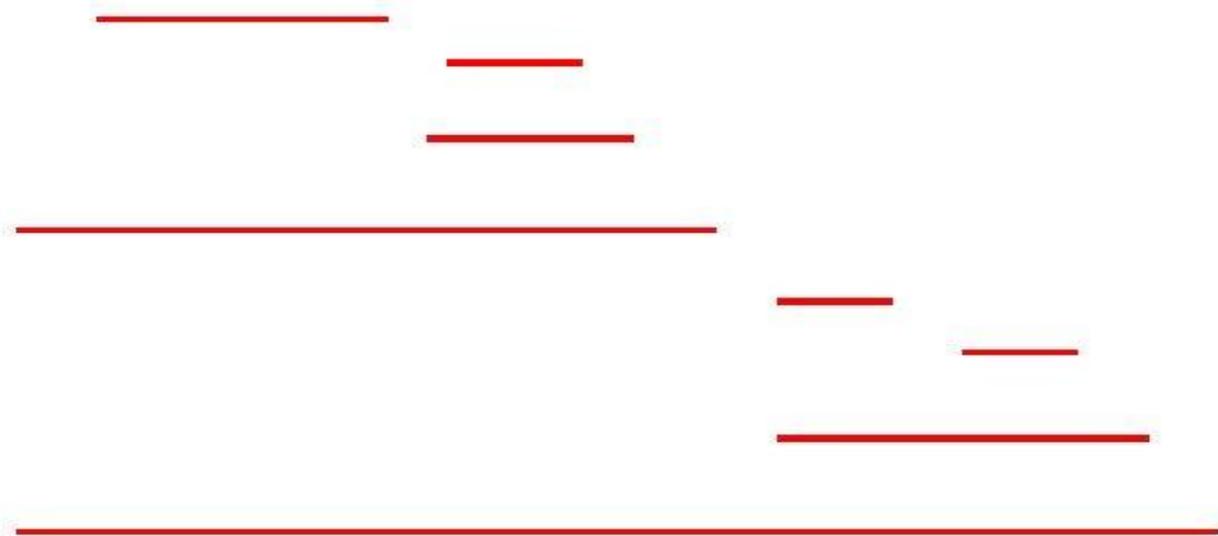
Left-first traversal;
Third-visit output;

10 22 3 4 x - 2 2 2 x - / + 14 2 - 1 3 + / x

第3步：栈操作

$$(10 + ((22 - 3 \times 4) / 2 - 2 \times 2) \times ((14 - 2) / (1 + 3)))$$

10 22 3 4 × - 2 2 2 × - / + 14 2 - 1 3 + / ×



Once a person has understood the way variables are used in programming, he has understood the quintessence of programming.

E. W. Dijkstra

结束语

- 数据、数据类型和变量分别表示什么？三者什么关系？
- 什么是数据结构？数据类型和数据结构什么关系？
- 为什么高级程序设计语言要提供“高级”数据结构？
- 程序员能否自行定义“数据类型”？

开放讨论题

- 请解读union数据结构：谈谈你对这个结构的认识，举例说明它的用途和使用方法
- 尝试证明second visit traversal 算法的正确性（部分正确性）