
计算机问题求解 – 论题2-11

- 基本的数据结构

2014年05月20日

自学检查

What is the key of an object in a dynamic set?

What is the satellite data of an object in a dynamic set?

LIST-DELETE(L, x)

1 **if** $x.prev \neq \text{NIL}$

2 $x.prev.next = x.next$

3 **else** $L.head = x.next$

4 **if** $x.next \neq \text{NIL}$

5 $x.next.prev = x.prev$



问题1:

全校学生分布在整个操场上，
怎么样就算是有“结构”
了？

问题2:

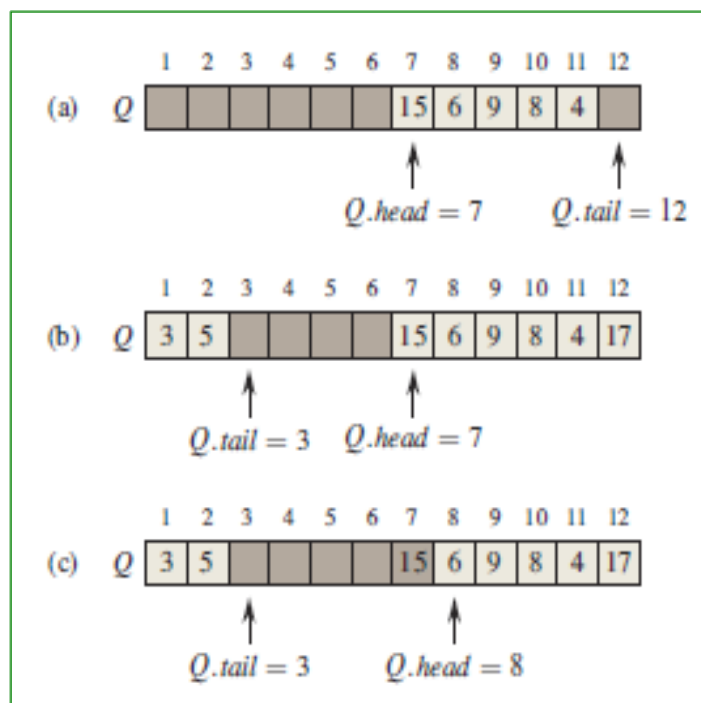
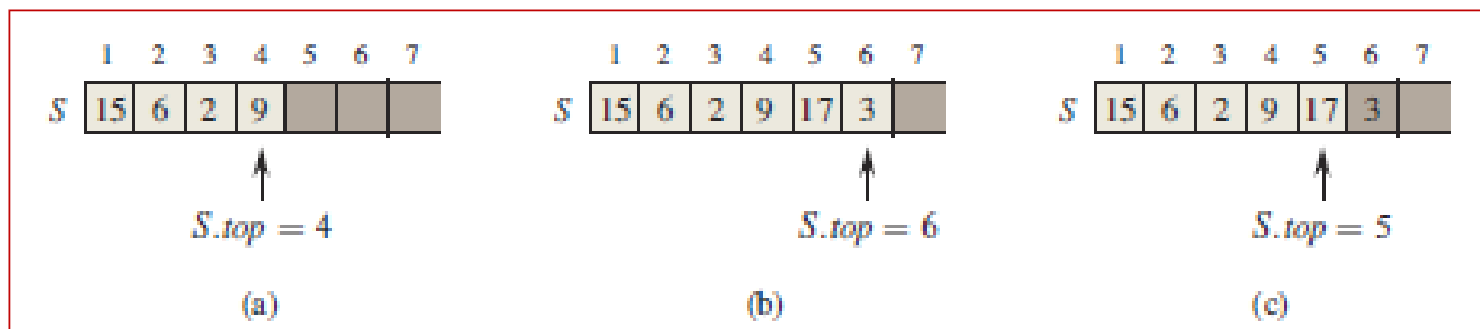
你能够说说在你用过的程序设计语言中有什么东西体现了数据的结构？

问题3:

仍然以全校的学生为例，你能够讨论一下“结构”和物理位置是什么关系？

问题4:

你认为队列与栈与其元素的物理位置分布有关吗?



问题5:

你根据什么区分这两个图各自描述哪种结构?

抽象数据类型的概念 – 以队列为例

Queue **create**()

Precondition: none.

Postconditions: If $q = \text{create}()$, then: q refers to a newly created object
 $\text{isEmpty}(q) = \text{true}$;

Boolean **isEmpty**(Queue q)

Precondition: none.

Object **front**(Queue q)

Precondition: $\text{isEmpty}(q) = \text{false}$;

void **enqueue**(Queue q , Object e)

Precondition: none.

Postconditions:

If $\text{isEmpty}(/q/) = \text{true}$, $\text{front}(q) = e$;

If $\text{isEmpty}(/q/) = \text{false}$, $\text{front}(q) = \text{front}(/q/)$;

$\text{isEmpty}(q) = \text{false}$;

void **dequeue**(Queue q)

Precondition: $\text{isEmpty}(q) = \text{false}$;

Postconditions: (精确描述需要特殊手段)

注意:

整个描述中完全不涉及如何“实现”，甚至不涉及“指针”。



问题 6 :

堆栈与队列的本质差别是什么？为什么我们需要这种差别？

问题8:

你觉得在我们的讨论的“结构”的背后是否有什么基本的数学概念吗？

从集合到“动态集合”

SEARCH(S, k)

A query that, given a set S and a key value k , returns a pointer x to an element in S such that $x.key = k$, or NIL if no such element belongs to S .

INSERT(S, x)

A modifying operation that augments the set S with the element pointed to by x . We usually assume that any attributes in element x needed by the set implementation have already been initialized.

DELETE(S, x)

A modifying operation that, given a pointer x to an element in the set S , removes x from S . (Note that this operation takes a pointer to an element x , not a key value.)

MINIMUM(S)

A query on a totally ordered set S that returns a pointer to the element of S with the smallest key.

MAXIMUM(S)

A query on a totally ordered set S that returns a pointer to the element of S with the largest key.

SUCCESSOR(S, x)

A query that, given an element x whose key is from a totally ordered set S , returns a pointer to the next larger element in S , or NIL if x is the maximum element.

PREDECESSOR(S, x)

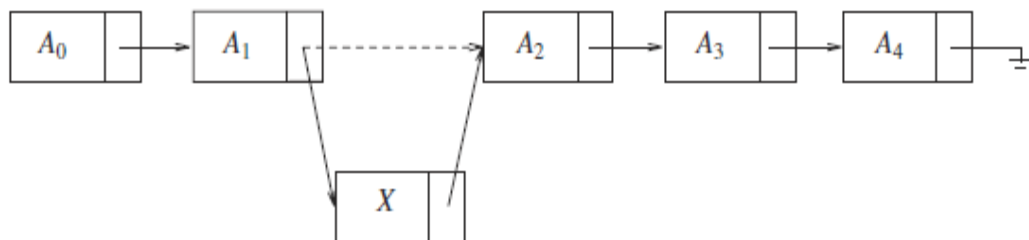
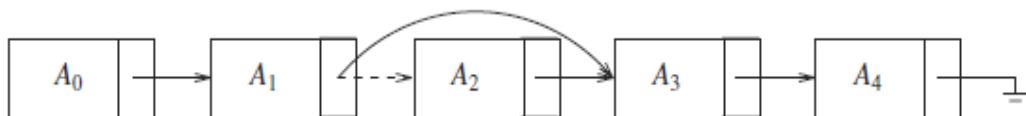
A query that, given an element x whose key is from a totally ordered set S , returns a pointer to the next smaller element in S , or NIL if x is the minimum element.

问题
结构
体现在那里?

问题10:

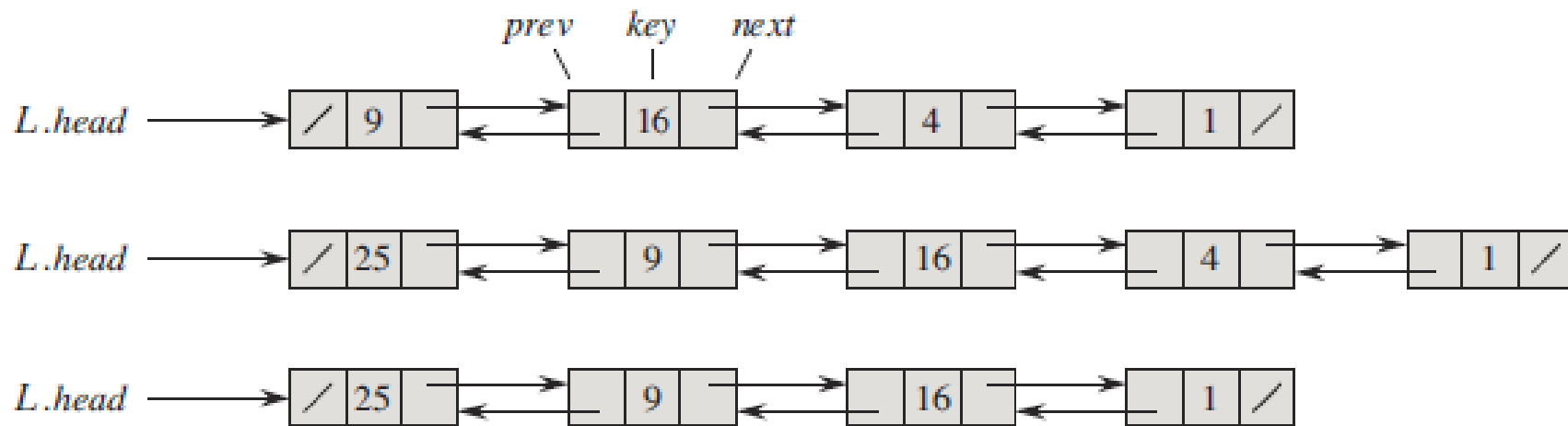
“实现” (implementation)
数据结构是什么意思？“指针”在数据结构的实现中有什么作用？

链表：一切操作均围绕指针



问题11:

这个结构有什么不便之处吗?



LIST-INSERT (L, x)

- 1 $x.next = L.head$
- 2 if $L.head \neq \text{NIL}$
- 3 $L.head.prev = x$
- 4 $L.head = x$
- 5 $x.prev = \text{NIL}$

LIST-DELETE (L, x)

- 1 if $x.prev \neq \text{NIL}$
- 2 $x.prev.next = x.next$
- 3 else $L.head = x.next$
- 4 if $x.next \neq \text{NIL}$
- 5 $x.next.prev = x.prev$

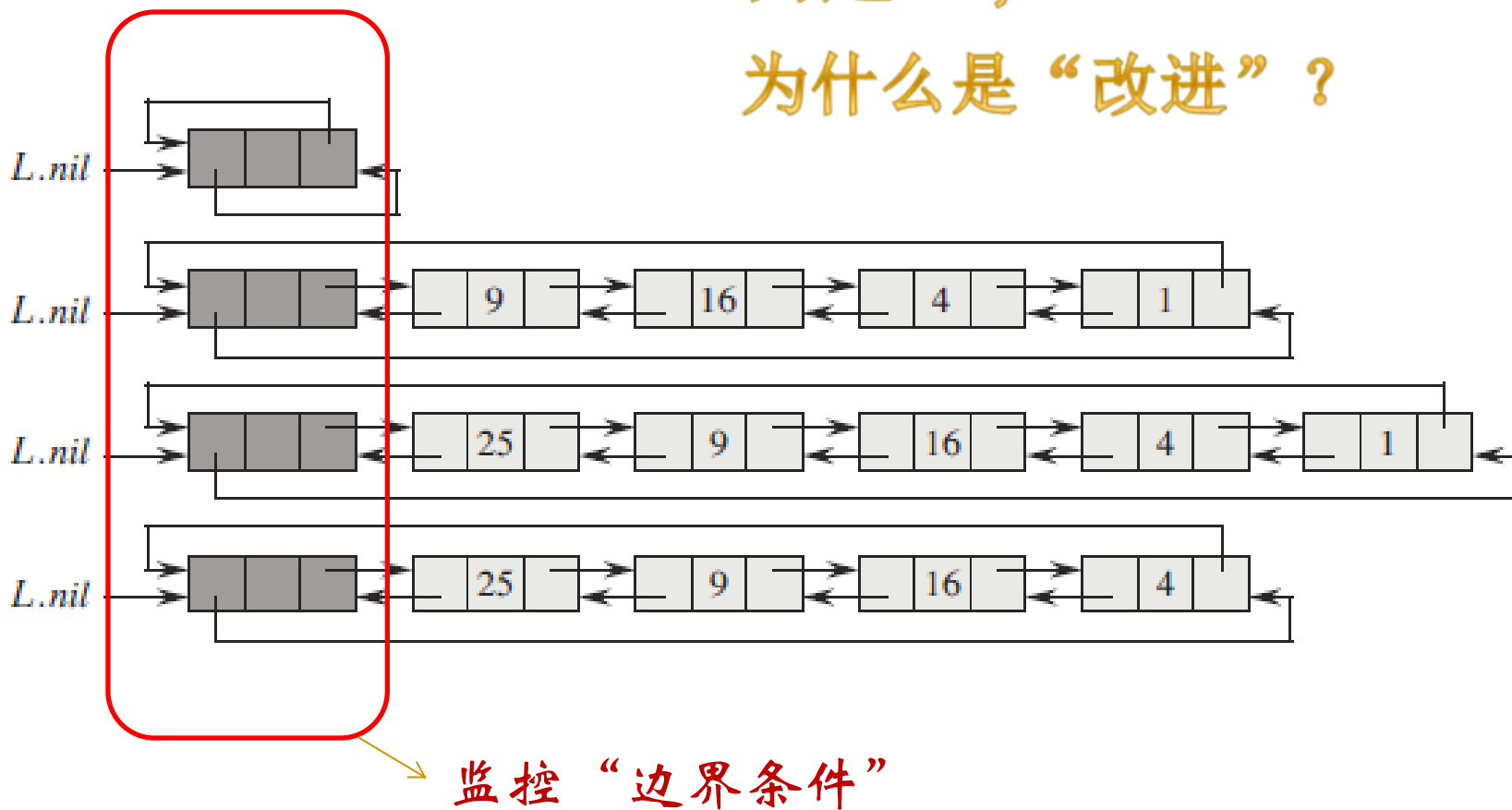
问题12:

删除一个对象的代价是多少?

进一步的改进

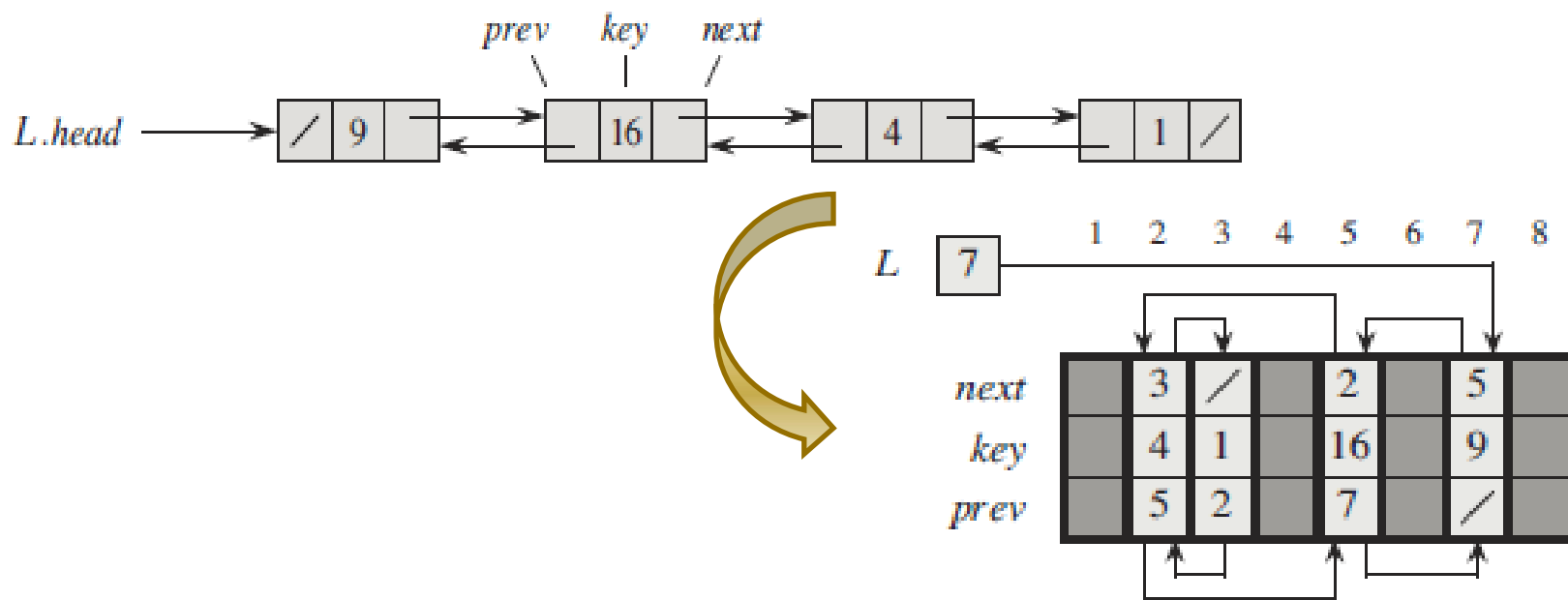
问题13;

为什么是“改进”？



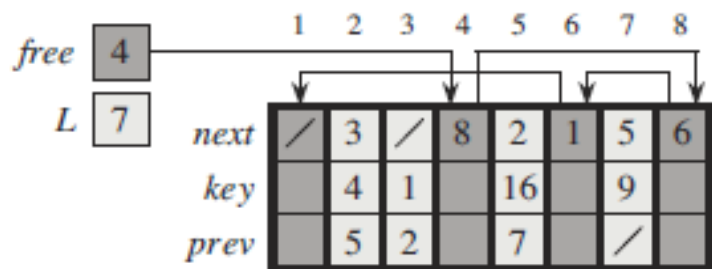
“此指针”未必是“彼指针”

- 通常我们熟悉的第一个“指针”是某种程序设计语言中提供的“语言设施”。
- 在用“指针”方法实现一种数据结构时，所谓的“指针”未必是语言中的指针，可以是实现其目的的其他手段。

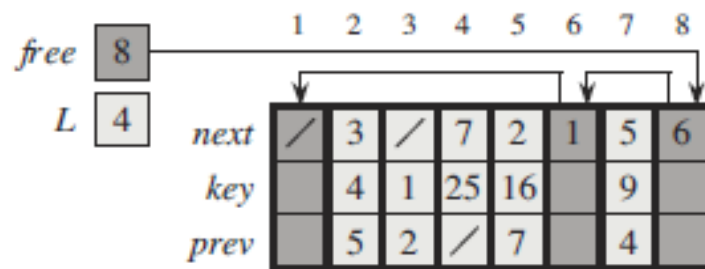


动态结构的存储管理

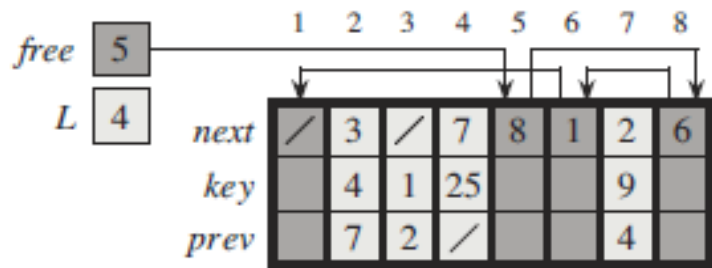
- 基本原理：在一个数组空间维护两个链表，“实际数据”和“可用空间”。“此消彼长”



(a)



(b)



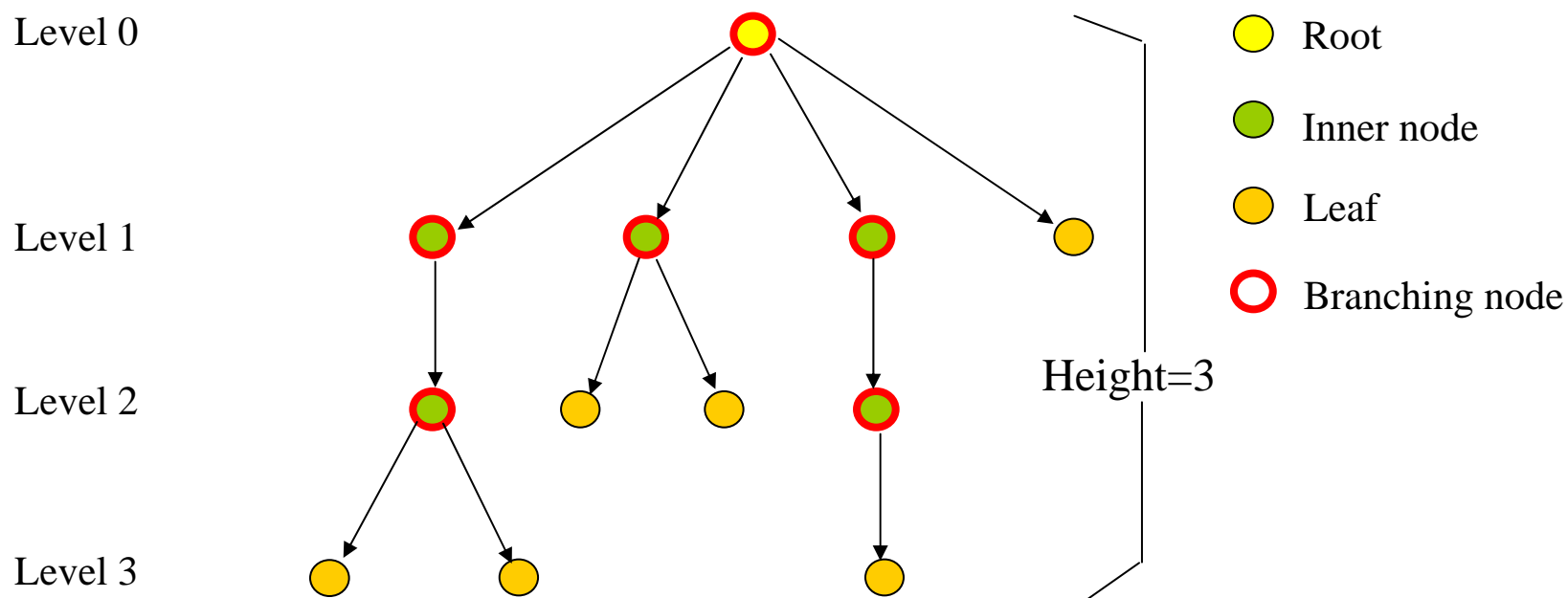
(c)

问题14:

你能解释一下
这几个图吗?

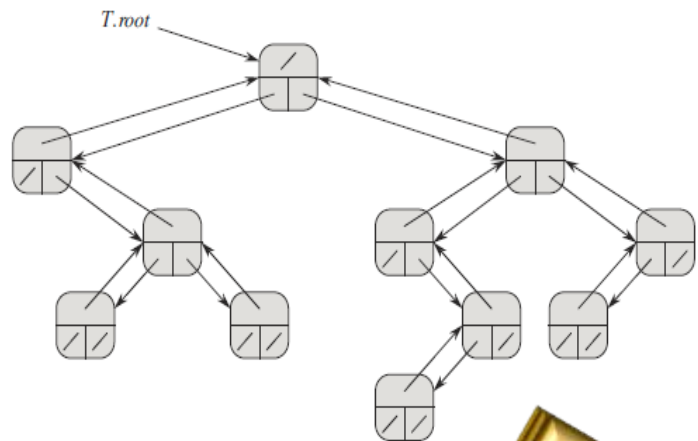
顺便问一下：
书上图10.8是什
么意思?

根树: 如何从链表的角度看它?



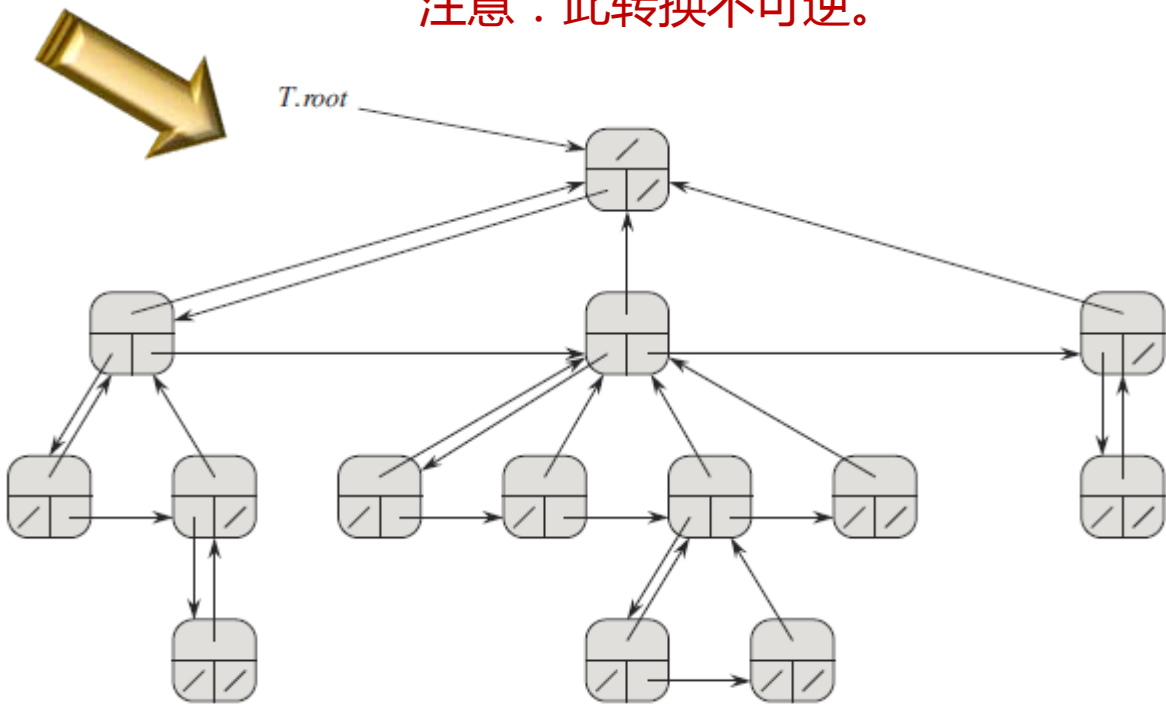
如果任意结点最多有两个子结点, 则该根树成为二叉树(binary tree), 显然用指针实现链表的方法很容易扩展到二叉树, 但是...

问题15:
“但是”什么?



这其实意味着：
即使问题逻辑需要多叉树，我们
也可以将其转换为二叉树来实
现。

注意：此转换不可逆。



课外作业

- TC pp.235-: ex.10.1-4; 10.1-5; 10.1-6
- TC pp.240-: ex.10.2-1; 10.2-2; 10.2-3; 10.2-6;
- TC pp.245-: ex.10.3-4; 10.3-5;
- TC pp.248-: ex.10.4-2; 10.4-3; 10.4-4
- TC pp.249-: prob.10-3