计算机问题求解 – 论题2–8

- 排序与选择的应用
- 基本数据结构

课程研讨

- TC第10章

排序和选择

- 排序（Sorting）
  - 冒泡、选择、插入
  - 快速排序
  - 归并排序
  - 堆排序

- 选择（Selection）
  - 最大/最小值、第二大值
  - 中位数/第k大值

Weighted Median

- For n distinct elements $x_1, x_2, \ldots, x_n$
  - Positive weights $w_1, w_2, \ldots, w_n$
  - $\sum_{i=1}^{n} w_i = 1$
- Weighted (lower) median: x_k satisfying
  - $\sum_{x_i < x_k} w_i < \frac{1}{2}$ and
  - $\sum_{x_i > x_k} w_i \leq \frac{1}{2}$

Weighted Median in $O(n\lg n)$?

- Sort n elements in increasing order by value of x_i, in $\Theta(n\lg n)$ time
- Accumulate the weights of elements from x_1 to x_k, until the aggregate weights exceeds ½
- The last element x_k is the weighted median
  - Verify by definition

Linear Worst-case Time ?

- $n < 2$, trivial case
- Find Actual median x_k and PARTITION
- Compute the aggregate weights of "smaller" and "larger" parts
  - If aggregate weights < ½: return x_k
  - Else reduce the problem to the larger part

```
WEIGHTED-MEDIAN(X)
if n = 1
  then return x_1
elseif n = 2
  then if w_1 ≥ w_2
         then return x_1
         else return x_2
else
       find the median x_k of X = {x_1, x_2, ..., x_n}
       partition the set X around x_k
       compute W_L = Σ_{x_i<x_k} w_i and W_G = Σ_{x_i>x_k} w_i
       if W_L < 1/2 and W_G < 1/2
         then return x_k
       elseif W_L > 1/2
         then w_k ← w_k + W_G
              X' ← {x_i ∈ X : x_i ≤ x_k}
              return WEIGHTED-MEDIAN(X')
       else w_k ← w_k + W_L
              X' ← {x_i ∈ X : x_i ≥ x_k}
              return WEIGHTED-MEDIAN(X')
```

Analysis

- $T(n) = T(n/2+1) + \Theta(n)$
  - Why the problem size is reduced by half?
- $T(n) = \Theta(n)$
  - by master's theorem

$E = \lg(b) \,/\, \lg(c) = 0,\ f(n) = \Theta(n)$

case 3: $f(n) \in \Omega(n^{E+\varepsilon})$, $(\varepsilon > 0)$, and $f(n) \in O(n^{E+\delta})$, $(\delta \geq \varepsilon)$, then:

$$T(n) \in \Theta(f(n))$$

*k*th Largest Element in Two Arrays

- Given two sorted arrays with n and m elements respectively, design an algorithm to find the kth largest element in the totally $(m+n)$ elements in $O(\log m + \log n)$ and explain the time complexity.

赛马问题

- 共有25匹马，每次可选5匹马进行比赛，并得到次序（无法计时）。问至少要比赛多少次才能确定跑得最快、次快和第三快的三匹马，并证明其最优性。

- 提示
  - 类比于寻找最大值、最小值、第二大值
  - 信息单元

分治法（Divide and Conquer）

```
solve(I)
    n=size(I);
    if (n≤smallSize)
        solution=directlySolve(I)
    else
        divide I into I1,… Ik;
        for each i∈{1,…,k}
            Si=solve(Ii);
        solution=combine(S1 ,… ,Sk);
    return solution
```

$T(n)=B(n)$ for $n \le$ smallSize

$$T(n)=D(n)+\sum_{i=1}^{k} T(size(I_i))+C(n)$$

for $n >$ smallSize

最小未出现自然数

- *n*个大小各不相同的自然数，找出不在这个自然数序列中出现的最小自然数。
  - “1、2、4、5”中最小未出现是3。

- 分别就下面两种情况设计算法
  - 若*n*个元素是已排序的
  - 若*n*个元素是未排序的

Finding the "Heavy" Element

- Find the element i with $freq(i) > n/2$ in an array of n elements. Here, $freq(i)$ is defined as the number of occurrence of i in the array.

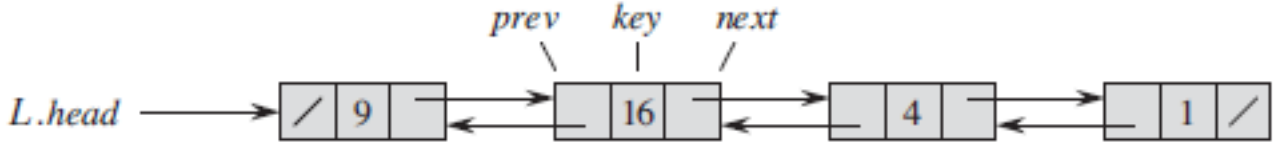
问题1：动态集合（dynamic set）

- 什么是动态集合（dynamic set）？
- 什么是字典（dictionary）？
- 你能解释动态集合的这些操作吗？
  - 哪些是查询（query），哪些是修改（modification）？

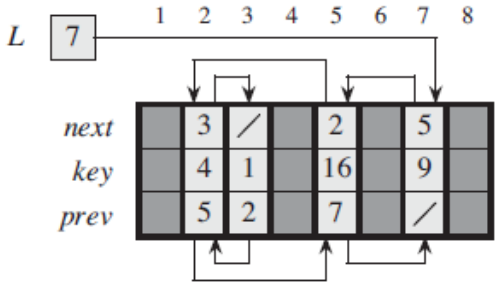
Search
Insert
Delete
Minimum
Maximum
Successor
Predecessor

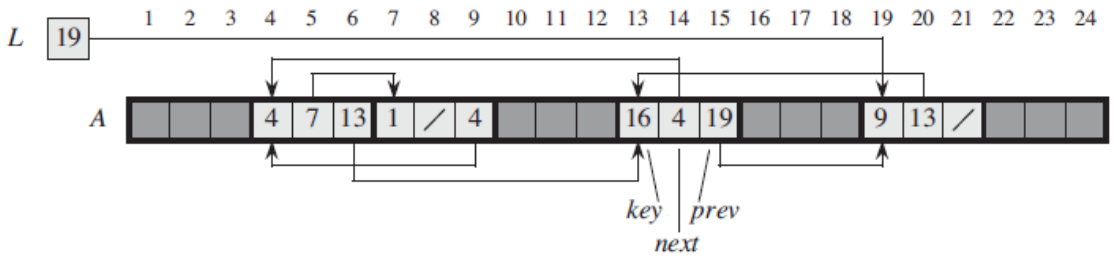
问题2-1：链表（linked list）

- 什么是单向/双向/循环链表？



- 你能正确写出双向链表的插入/删除操作吗？
- 你能正确写出单向链表的插入/删除操作吗？
- 你能比较链表的两种数组实现的优劣吗？

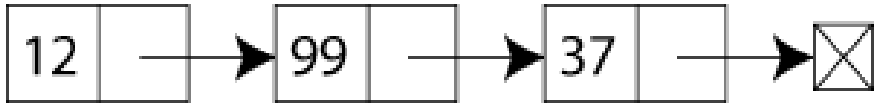




问题2-2：链表实现动态集合

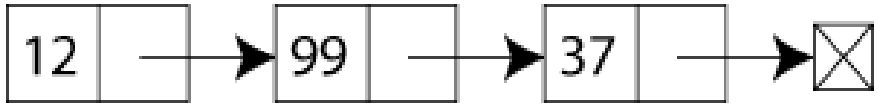
- 你能利用链表实现动态集合的所有操作吗？
  它们的运行时间分别是多少？

Search
Insert
Delete
Minimum
Maximum
Successor
Predecessor



问题2-3：链表操作

- 你能写出一个算法，将单向链表反转吗？

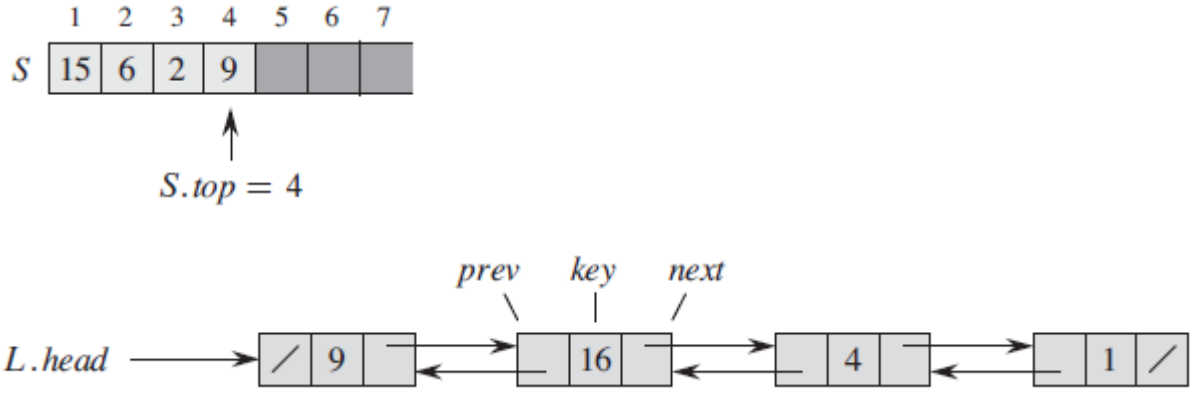


问题2-4：链表操作

- 有两个单向链表，如何快速找到其公共的节点?

问题3-1：栈（stack）

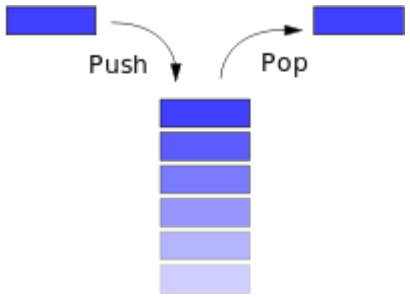
- 什么是栈？
- 你能正确写出栈的数组实现的push/pop操作吗？
- 你能用链表来实现栈吗？



问题3-2：栈实现动态集合

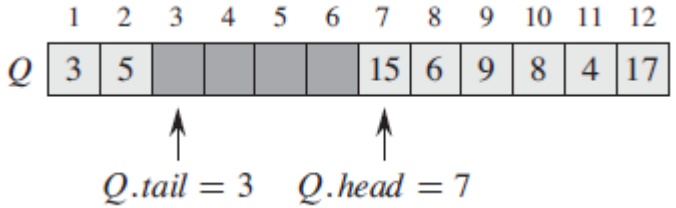
- 你能利用栈实现动态集合的所有操作吗？它们的运行时间分别是多少？

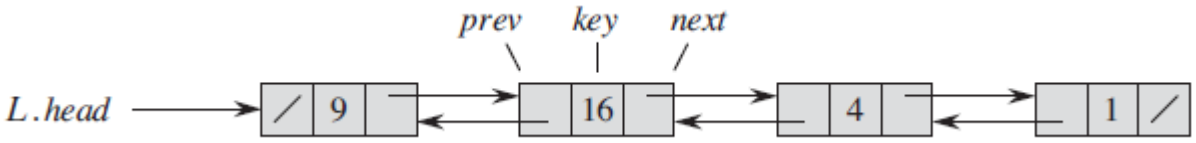
Search
Insert
Delete
Minimum
Maximum
Successor
Predecessor



问题4-1：队列（queue）

- 什么是队列？
- 你能正确写出队列的数组实现的enqueue/dequeue操作吗？
- 你能用链表来实现队列吗？

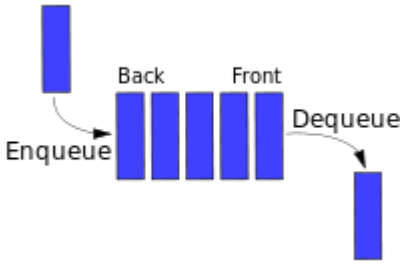




问题4-2：队列实现动态集合

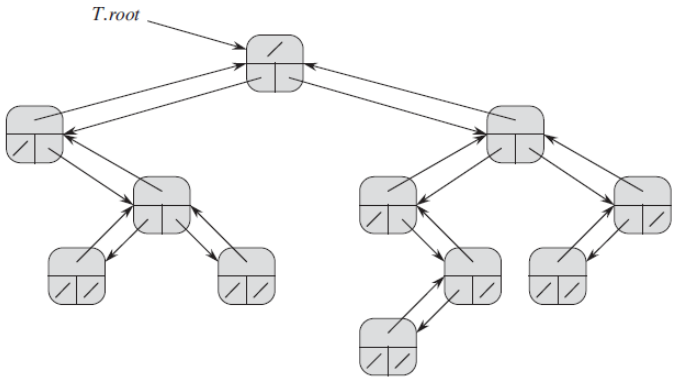
- 你能利用队列实现动态集合的所有操作吗？它们的运行时间分别是多少？

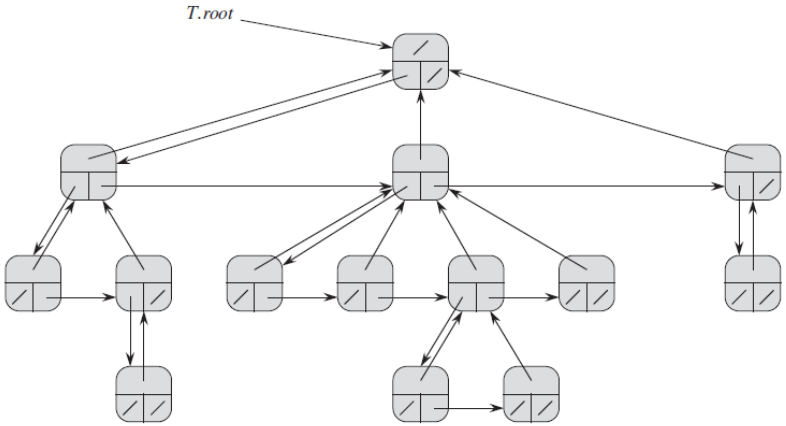
Search
Insert
Delete
Minimum
Maximum
Successor
Predecessor



问题5-1：树（rooted tree）

- 什么是树？
- 二叉树和链表有什么异同？
- 什么是left-child, right-sibling representation？
- 你能用数组来实现left-child, right-sibling representation吗？

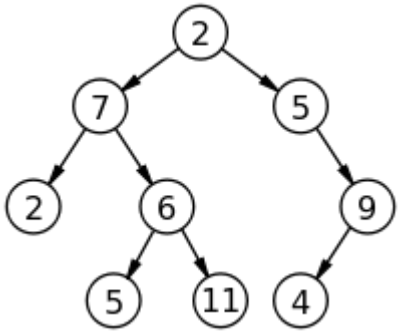




问题5-2：二叉树实现动态集合

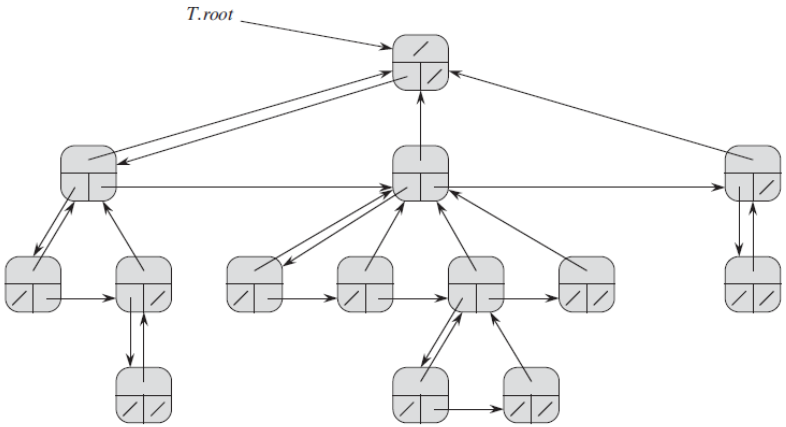
- 你能利用二叉树实现动态集合的所有操作吗？它们的运行时间分别是多少？

Search
Insert
Delete
Minimum
Maximum
Successor
Predecessor



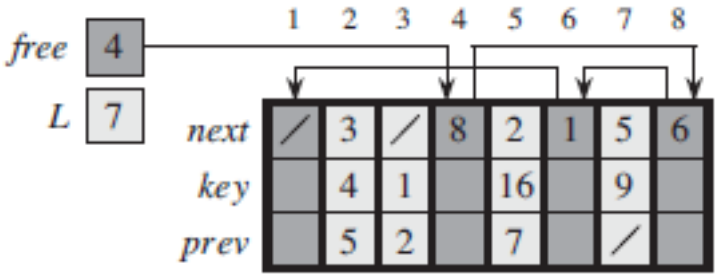
问题5-3： rooted tree (续)

- The left-child, right-sibling representation of an arbitrary rooted tree uses three pointers in each node: *left-child*, *right-sibling*, and *parent*. From any node, its parent can be reached and identified in constant time and all its children can be reached and identified in time linear in the number of children. Show how to use only two pointers and one boolean value in each node so that the parent of a node or all of its children can be reached and identified in time linear in the number of children.



问题6：申请和释放对象

- 为什么需要申请和释放对象？
- “free list”本质上是一种什么样的数据结构？



- 你怎么理解“service several linked lists with a single free list”？
- 你觉得这种做法有什么优缺点？