

# 算法设计与分析习题

## Phase IV: 图遍历

- 4.2, 4.5, 4.7, 4.8, 4.9, 4.12, 4.13, 4.14, 4.16, 4.17, 4.18, 4.20, 4.22, 4.23
- 5.1, 5.2, 5.4, 5.8, 5.9, 5.10

## Phase III: 选择与查找

- 8.2, 8.4, 8.5, 8.6, 8.8, 8.9
- 9.4, 9.6, 9.8, 9.12

☐ 1 (不同哈希表的存储效率) 哈希表 $H$ 在闭哈希(*closed-address hashing*)方法下是一个链表头组成的数组, 而在开哈希(*open-address hashing*)方法下是一个关键字的数组。假设一个关键字需要一个单位的存储, 而一个链表节点需要两个单位的存储, 其中一个存放关键字, 另一个存放链表节点的指针。考虑在闭散列下如下的负载因子:  $0.25, 0.5, 1.0, 2.0$ 。设 $h_C$ 是使用闭哈希时哈希表的大小(即链表头数组初始时有多少空位)。

1. 请计算闭哈希表的空间消耗。假设同样的空间用于开散列表, 它的负载因子会是多少?
2. 假设一个关键字要占据4个单位的存储, 而一个表节点需要5个单位的存储(4个用于关键字, 一个用于指针), 请再次计算上一问中的问题。

☐ 2 (平摊分析) 现在有一个由数组组成的集合, 数组 $i$ 的大小是 $2^i$ 。每个数组不是空(没有元素)就是满的(填满元素)。例如, 11 个元素存储在下面的数组里:

$A_0 : [a_1]$   
 $A_1 : [a_2, a_3]$   
 $A_2 : \text{empty}$   
 $A_3 : [a_4, a_5, \dots, a_{11}]$

现在插入一个新的元素, 称作 $a_{12}$ 。首先创建一个新的大小为1的数组存放 $a_{12}$ , 现在我们查看 $A_0$  是否为空, 如果 $A_0$ 为空, 那么就令这个新数组成为 $A_0$ ; 如果不为空(如上面的例子), 就将这个新数组和 $A_0$ 合并为一个新的数组(在上面的例子中 $A_0$ 就变为 $[a_1, a_{12}]$ ), 并且再继续查看 $A_1$ 是否为空, 如

果 $A1$ 为空,就令这个新的 $A0$ 成为 $A1$ ;如果不为空,则将其和 $A1$ 合并,并且继续查看 $A2$ ,反复执行上述操作。所以,在上面的例子中插入 $a_{12}$ ,我们最终得到的新的数组的集合是:

$A0 : \text{empty}$

$A1 : \text{empty}$

$A2 : [a_1, a_2, a_3, a_{12}]$

$A3 : [a_4, a_5, \dots, a_{11}]$

现在我们另创建一个新的大小为 $1$ 的数组的开销是 $1$ ,合并两个大小分别为 $m$ 的数组的开销为 $2m$ ,所以上面的例子所需要的总开销是 $1+2+4$ 。请用平摊分析的方法分析插入操作的复杂度。

**☐ 3** 使用两个后进先出的栈可以实现一个先进先出的队列。假设有三个操作, *push* (压栈)、*pop* (出栈) 和 *empty* (判断是否为空), 每个操作的代价都是 $1$ 。

- 请用上述栈的操作实现一个队列。
- 利用平摊分析法分析你所提算法的时间复杂度。

## Phase II: 排序

- 6.8, 6.9, 6.10, 6.13, 6.15
- 7.1, 7.2, 7.3, 7.4, 7.6

## Phase I: 准备知识

- 1.1, 1.2, 1.3, 1.5, 1.6, 1.7
- 2.2, 2.3, 2.5, 2.6, 2.7, 2.15, 2.17, 2.18, 2.20
- 3.2, 3.4, 3.5, 3.6, 3.7