3.3 查找

Problem 3.3.1
如果数组中的第1个元素到第i个元素值是单调递增的，而从第i个元素到最后一个元素值是单调递减的，我们称其为“ 单峰”数组。现在给定一个大小为n的“ 单峰”数组E，请设计复杂度为$O(logn)$的找出E 中最大元素的算法。

Problem 3.3.2
给定一个有n个互不相同的有序整数$\{a_1, a_2, \cdots, a_n\}$的序列。请设计算法判断是否存在某个下标$i$满足$a_i = i$。例如，$\{-10, -3, 3, 5, 7\}$，有$a_3 = 3$；在$\{2, 3, 4, 5, 6, 7\}$ 不存在满足条件的i。

Problem 3.3.3
数组$A[1..n]$存放的是n个有序的互不相同的正整数。请设计算法找出A中缺少的最小的正整数。例如，$A = [1, 2, 4, 5]$，缺少的最小的正整数是3。

Problem 3.3.4
散列表H的类型在闭散列方法下是一个表的引用数组，而在开散列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内存，而一个连接表节点需要两个单位的内存，其中一个存放关键字，另一个存放表的引用。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是闭散列下散列单元的数量。

1. 给出总体的空间需求，包括在闭散列下的表所需要的空间。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个单位的内存，而一个表节点需要5个单位内存（4个用于关键字，一个用于余下表的引用），重复上一问中的问题。

Problem 3.3.5
对下面每个问题给出符合要求的算法，可以在你的算法中使用已有的排序算法。

(a) S是由n个整数组成的数组，并且未排序。请给出算法用来找到整数对$x, y \in S$并且$|x - y|$ 最大，最坏情况下，你的算法的复杂度为$O(n)$。

(b) S是由n个整数组成的有序数组。请给出算法用来找到整数对$x, y \in S$并且$|x - y|$最大，最坏情况下，你的算法的复杂度为$O(1)$。

(c) S是由n个整数组成的数组，并且未排序。请给出算法用来找到整数对$x, y \in S$并且$|x - y|(x \neq y)$最小，最坏情况下，你的算法的复杂度为$O(nlogn)$。

(d) S是由n个整数组成的有序数组。请给出算法用来找到整数对$x, y \in S$并且$|x - y|(x \neq y)$ 最小，最坏情况下，你的算法的复杂度为$O(n)$。

Problem 3.3.6
令M是一个$m \times n$的矩阵，矩阵中的每一行元素从左到右按照升序排列，每一列的元素从上到下按照升序排列。请给出高效的算法在矩阵中查找给定的元素x（x可能不存在），并给出最坏情况下所需要的比较次数。

Problem 3.3.7
给定由n个实数组成的集合S和一个实数值a。请给出算法判定在S中是否存在两个元素的和为a。

(a) 假设S是未排序的。给出复杂度为$O(nlogn)$的算法。

(b) 假设S是排完序的。给出复杂度为$O(n)$的算法。

Problem 3.3.8
下面两个问题都是求集合A, B的并集，其中$n = max\{|A|, |B|\}$。

(a) 假设集合A, B中的元素都是乱序的。请给出复杂度为$O(nlogn)$的算法。

(b) 假设集合A, B中的元素都是排好序的。请给出复杂度为$O(n)$的算法。

Problem 3.3.9
任意的一棵RB_h或者ARB_h的black height是良定义的，为h。现在另T为一棵RB_h，A为一棵ARB_h，请证明下列命题：

(a) T至少有$2^h - 1$个内部黑色节点。

(b) T至多有$4^h - 1$个内部节点。

(c) 任意黑色节点的深度至多是它的black depth的两倍。

(d) A至少有$2^h - 2$个内部黑色节点。

(e) A至多有$(4^h)/2 - 1$个内部节点。

(f) 任意黑色节点的深度至多是它的black depth的两倍。

Problem 3.3.10
证明：对于一个并查集程序，长度为m，一组中有n个元素，那么它如果使用无权值的合并和直接查找的方法来实现的话，需要最多$n + mn$次读写父节点信息操作（link operation）。

Problem 3.3.11
说明一个大小为m的并查集需要$\Omega(n + (m - n)logn)$的时间来进行查找和合并，如果查找使用直接查找（无路径压缩），合并使用加权值的合并（wUnion）。

Problem 3.3.12
S_k树是如下定义的：

- S_0如果只有一个节点，它是一棵树。
- 对于$k > 0$，一个S_k是由两棵不相交的S_{k-1}树构成，通过将其中一棵的根连接到另一棵的根上[2]。

证明：如果T是一棵S_k树，则T有2^k个顶点，高度为k，并且在第k层有唯一的顶点，这个在第k层的顶点称为S_k树的柄。

Problem 3.3.13
使用练习3.3.12中的定义和结果，证明S_k树的以下性质：设T是一棵S_k树，其树柄为v。有不相交的树T_0，T_1，$\cdots$，T_{k-1}，不包含v，它们的根分别是r_0，r_1，$\cdots$，r_{k-1}，这样，

[2]请参考Sara Baase and Allen Van Gelder. Computer Algorithms-Introduction to Design and Analysis(算法设计与分析)图6.28

1. T_i是一棵S_i树，其中0 ≤ i ≤ k-1,且

2. T是通过将v加到r_0，r_i加到r_{i-1}上得到的，其中0 ≤ i ≤ k-1

这个分解过程在S_4的情形下请参考图6.29[3]。

Problem 3.3.14

使用在练习3.3.12中的定义和结果，证明S_k树的以下性质：设T是一棵S_k树，其根为r而树柄为v。有互不相交的树T_0'，T_1'，$\cdots$，T_{k-1}'，不包含r，它们的根分别是r_0'，r_1'，$\cdots$，r_{k-1}'，这样，

1. T_i'是一棵S_i树，0 ≤ i ≤ k-1，

2. T是通过如下方式构造的：即将每个r_i'添加到r上，对于所有的0 ≤ i ≤ k-1，

3. v是T_{k-1}'的树柄。

这一过程在树为S_4的时候请参考图6.30[4]。

Problem 3.3.15

采用闭地址散列的方式避免哈希表的冲突问题。现在假设有k个关键字需要插入该哈希表中（有n个槽），每个关键字等可能地插入每个槽中。求恰好有k 个关键字插入某个特定槽的概率。

Problem 3.3.16

以下是程序自动分析中的一个问题。对于一组变量$x_1, \cdots, x_n$，给定一些形如"$x_i = x_j$"的等式约束和形如"$x_i \neq x_j$"的不等式约束。这些约束是否能同时满足？

例如，如下一组约束：

$$x_1 = x_2, x_2 = x_3, x_3 = x_4, x_1 \neq x_4$$

是无法同时满足的。请给出一个有效的算法，判断关于n个变量的m个约束是否可同时满足。

Problem 3.3.17

给定一个由n个整数组成的集合S和一个整数T，请给出复杂度为$O(n^{k-1}logn)$的算法用来判定在S中是否存在k个整数的和为T。

Problem 3.3.18

请给出高效的算法在n个值之中找到第三大的，并回答最坏情况下所需要的比较次数是多少和你所给出的算法在执行过程中是否必须找到最大的和次大的元素。

Problem 3.3.19

假设有一个数组$A[1...n]$，它满足以下两个条件：$A[1] \geq A[2]$；$A[n-1] \leq A[n]$。当元素$A[x]$ 并不大于它的两个邻居（$A[x-1] \geq A[x], A[x+1] \geq A[x]$）时，我们称$A[x]$为局部最小。例如，在下面的数组中有六个局部最小元素。

9	7	7	2	1	3	7	5	4	7	3	3	4	8	6	9

[3]Sara Baase and Allen Van Gelder. Computer Algorithms-Introduction to Design and Analysis(算法设计与分析) 图6.29
[4]Sara Baase and Allen Van Gelder. Computer Algorithms-Introduction to Design and Analysis(算法设计与分析) 图6.30

(a) 我们可以用$O(n)$的时间通过扫描一遍数组找到一个局部最小元素。请设计一个用$O(logn)$的时间找到一个局部最小元素的算法。

(b) 在给定的边界条件下，请说明至少存在一个局部最小元素。

Problem 3.3.20
假设你只可以使用加法和移位操作。请给出一个高效的算法计算$\lceil\sqrt{N}\rceil$。

1. 给定一个自然数N有n个比特，请给出复杂度为$O(n^2)$的算法计算$\lceil\sqrt{N}\rceil$。

2. （选做）请将你的算法改进到$o(n^2)$。

Problem 3.3.21
数组A和B分别有n个整数。给定一个整数x，请设计算法找到是否存在$a \in A$和$b \in B$并且满足$x = a + b$，并分析算法的复杂度。

Problem 3.3.22
编写一个客户端用来创建一个符号表完成对字母表示的成绩和数值学分之间的映射，如下表所示，然后从标准输入一个字母成绩的列表，算出并输出对应的GPA（平均成绩）。

A+	A	A-	B+	B	B-	C+	C	C-	D	F
4.33	4.00	3.67	3.33	3.00	2.67	2.33	2.00	1.67	1.00	0.00

Problem 3.3.23
现在有一个由数组组成的集合，数组i的大小是$2i$。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：

$$\begin{aligned} A0 &: [a_1] \\ A1 &: [a_2, a_3] \\ A2 &: empty \\ A3 &: [a_4, a_5, \cdots, a_{11}] \end{aligned}$$

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$A0$ 是否为空，如果$A0$为空，那么就另这个新数组成为$A0$；如果不为空（如上面的例子），就将这个新数组和$A0$合并为一个新的数组（在上面的例子中$A0$就变为$[a_1, a_{12}]$），并且再继续查看$A1$是否为空，如果$A1$为空, 就另这个新的$A0$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：

$$\begin{aligned} A0 &: empty \\ A1 &: empty \\ A2 &: [a_1, a_2, a_3, a_{12}] \\ A3 &: [a_4, a_5, \cdots, a_{11}] \end{aligned}$$

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。

Problem 3.3.24

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。队列的两个操作可以按照如下方式实现：

- Enqueue(x)入队:将x元素压入栈1。如果栈1满，那么返回错误。
- Dequeue()出队:如果栈2为空，将栈1的元素依次出栈并压入栈2。然后，将栈2的元素出栈，并返回结果。如果栈2与栈1均为空，返回错误。

请回答下面两个问题：

(a) 解释这个方法的正确性。

(b) 利用平摊分析法分析该算法的时间复杂度。