3.2 选择

Problem 3.2.1

给出一个算法，使得在最坏情况下，它能够只利用六次比较来找出五个元素的中间值。描述该算法的步骤，但不需要写出代码。利用决策树图的形式给出你的算法。决策树是一棵二叉树，每个内部节点代表一次比较“$A[i]:A[j]$“，每个叶子节点代表的是数组中元素。

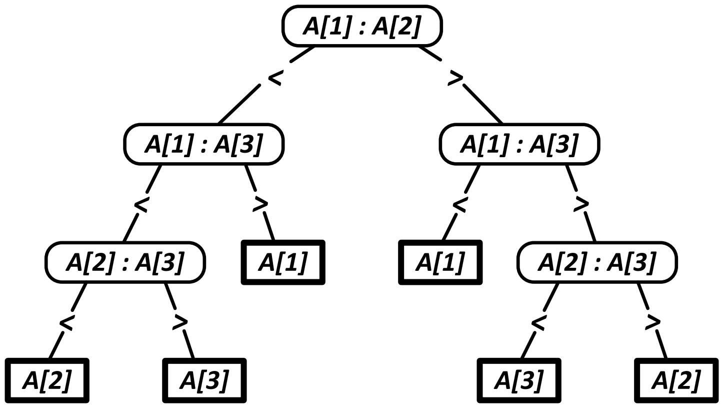

图 3.1: 至多用三次比较找出三个不同元素的中值

Problem 3.2.2

假设对一个含有n个元素的集合，某算法只用比较来确定第i小的元素。证明：无需另外的比较操作，它也能找到比第i小的元素小的$i-1$个元素和比第i小的元素大的$n-i$个元素。

Problem 3.2.3

假设已有一个用于选择中位数的“黑盒”算法$\mathcal{A}$，它在最坏情况下需要线性运行时间。请给出基于已有的黑盒算法$\mathcal{A}$，选择任意第 k 小元素的算法$\mathcal{B}$，要求算法$\mathcal{B}$最坏情况下也是线性时间的。

Problem 3.2.4 (k Largest Numbers)

给定有n个数的集合，现要求找出其中的前k大的k个数（得出的这k个数是排好序的，即知道哪个是第1 大，第2大，$\cdots$, 第k大），请设计三个基于比较的算法，并使得算法的最坏时间复杂度分别符合下面三个要求：

(a) $O(n \log n)$

(b) $O(n + k \log n)$

(c) $O(n + k \log k)$

Problem 3.2.5 (k^{th} element)

给定一个堆，其中有n个元素。请选出其中的第k大的元素。假设$k \ll n$，选择的代价要求是k的函数（与n无关）。

Problem 3.2.6

给定一个有n个不同正数的集合S，用M表示S的中值。请设计算法找出S中和M的大小最接近的k个数（k远小于n）。例如，集合$S=\{6,7,50,800,900\}$，中值M是50，两个（$k=2$）和中值M最接近的数是6和7。

(a) 给出时间复杂度为$O(n\log n+k)$的算法。

(b) 给出时间复杂度为$O(n+k\log k)$的算法。

Problem 3.2.7

(a) 给定两个有序数组$A[1...n],B[1...n]$和一个整数k，请给出算法用$O(logn)$的时间找到$A\bigcup B$中的第k 小的元素。如果$k=1$，算法得到的应当是$A\bigcup B$中最小的元素；如果$k=n$，算法得到的应当是$A\bigcup B$的中值。你可以假设数组中没有重复元素。（提示：先解决$k=n$这种特殊情况。）

(b) 给定三个有序数组$A[1...n],B[1...n],C[1...n]$和一个整数$k$，请给出算法用$O(logn)$的时间找到$A\bigcup B\bigcup C$ 中第k小的元素。

(c) 给定二维数组$A[1...m][1...n]$，数组的每一行都是有序的和一个整数k。请给出一个算法用尽可能短的时间找到数组中第k小的元素，并给出算法的时间复杂度。

Problem 3.2.8 (动态发现中值)

请设计一个数据结构支持对数时间复杂度的插入操作，常数时间发现中值，对数时间删除中值。（提示：利用1 个最小堆和1个最大堆。）

Problem 3.2.9

现有n个互不相同的数$x_1,x_2,\cdots,x_n$，每个数都有一个非负权重，$w_1,w_2,\cdots,w_n$，并且满足$\sum_{1\le i\le n}w_i=1$，其中被称之为“加权中位数”的数x_k必须满足不等式

$$\sum_{x_i<x_k}w_i<\frac{1}{2},\sum_{x_i>x_k}w_i\le\frac{1}{2}$$

(a) 证明当$w_i=\frac{1}{n}$（$i=1,2,\cdots,n$）时，$x_1,x_2,\cdots,x_n$的中值即“加权中位数” 。

(b) 请给出用基于排序的方法找出“加权中位数”的算法，并且最坏时间复杂度满足$O(nlogn)$。

(c) （选做）请给出最坏时间复杂度为$\Theta(n)$的找到“加权中位数”的算法。