第6章 变治法

张翰林 20201060351

**1. 考虑这样一个问题：它要找出个数字构成的一个数组中两个最接近数的距离（两个数和之间的距离定义为）。**

**（1）设计一个基于预排序的算法来解决该问题 并确定其效率类型。**

**（2）对该算法的效率类型和蛮力法的效率类型进行比较。**

（1）

**方案**：先使用任意排序算法进行预排序，得到有序数组。随后只需要遍历整个有序数组，依次对相邻元素作差，比较该差值是否更小即可。

**效率分析**：假设使用快速排序进行预排序，则该部分的平均时间复杂度为。随后遍历整个数组作差并比较，时间复杂度为，则总共时间复杂度为。

（2）

上述分析已知预排序方法的时间复杂度为，接下来分析蛮力法。

**方案**：首先作差求绝对值，即为距离。若输入规模为，则共有种组合，也就是共有个距离。随后需要逐个遍历，从中找出最小的值。

**效率分析**：作差需要执行次，则该部分时间复杂度为，比较部分需要遍历个结果，故该部分时间复杂度也为。故总共时间复杂度为。

二者比较后，可发现本题中使用预排序可以显著降低时间复杂度。

**2. 利用图示法给出对以下列表构造最大堆的过程：。**

|  |  |
| --- | --- |
| ①原本状态 |  |
| H=[23, 17, 21, 3, 42, 9, 13, 1, 2, 7, 35, 4] |
| ② | 图示, 示意图  描述已自动生成 |
| H=[23, 42, 21, 3, 17, 9, 13, 1, 2, 7, 35, 4] |
| ③ | 图示, 示意图  描述已自动生成 |
| H=[23, 42, 21, 3, 35, 9, 13, 1, 2, 7, 17, 4] |
| ④ | 图示, 形状  描述已自动生成 |
| H=[42, 23, 21, 3, 35, 9, 13, 1, 2, 7, 17, 4] |
| ⑤ | 图示  描述已自动生成 |
| H=[42, 35, 21, 3, 23, 9, 13, 1, 2, 7, 17, 4] |

至此，最大堆构造完成。

**3. 给定列表，为其构造一个堆。**

**（1）sift-down操作将执行多少次？为什么？**

**（2）给出一个具有最少元素比较次数的例子，给出其中比较操作执行的次数，并进行解释说明。**

（1）

执行9次。因为该列表满足，所以对于任意父结点，其子结点和的值必然大于，所以所有的父结点都要执行操作。而父结点是前个，即，故执行9次。

（2）

根据自底向上构造堆的伪代码可知，比较操作全部在循环即操作内，所以只需每次进入后都可以进行到，循环就只执行1次，进而比较次数最少。容易知，执行最少的情况是原列表就是一个堆的情况。

所以例子为,为一个标准的堆的情况。

根据上述情况，每个父结点都会进入1次循环，每个循环中有3个判断，故每个父结点进行2或3次判断（取决于是否有2个子结点）。父结点共有9个，且最后的父结点也有两个子结点，故进行次比较操作。