《算法设计与分析》课程作业一

尹达恒

2020/10/14

1 快速排序

1.1 问题描述

Description

给定一维 int 型数组 a[0,1,...,n-1]，使用快速排序方法，对其进行从小到大排序，请输出递归过程中自顶自下第二层的划分结果，其中最顶层为第一层，即最终的排序结果层。

划分时请用第 1 个元素作为划分基准，并使用课件上的方法进行一次扫描实现划分。

Input

输入第 1 行有一个 int 型正整数 m(m<100)，表示有 m 行输入。每行输入的第一个数为 int 型正整数 n(8<n<1000)，后面接着输入 n 个 int 型整数。

Output

对每组数据, 输出自顶自下第二层的划分结果。

Sample Input

```
2
11 6 3 7 8 5 1 4 2 4 9 10
12 6 3 7 8 4 5 1 11 2 4 9 10
```

Sample Output

```
2 3 1 4 4 5 6 7 8 9 10
2 3 1 4 4 5 6 10 8 7 9 11
```

1.2 算法思路

1. 划分：以数组中的第一个元素为基准值，从数组中的第二个元素开始扫描，比基准值小的放右边，比基准值大的放左边。
2. 处理：
   - 初始时基准坐标 p= 数组开头位置；

- i 从数组第二个元素开始遍历，若 i 位置的值大于基准值，则与 p 位置后一位的值交换，并令 p 自增；
- 最后令数组开头的值与 p 位置的值交换。

3. 递归：使用同样的方法递归地处理左边和右边的子数组。
4. 输出第二层结果：设置一个层标记 l=1，每次递归都将此值加 1 后作为参数传入，递归函数内若检测到 l>2 则退出并输出结果。

Algorithm 1: 快速排序算法伪代码

```
1  Function QuickSort(S, n) begin
       Input: 未排序的数组 S、数组长度 n、递归层数 l
       Output: 排好序的数组 S
2      if (l ≤ 2) ∧ (n > 1) then
3          p = 0;
4          for i ∈ {1, 2, 3, ..., n} do
5              if S_0 > S_i then
6                  交换 S_{p+1} 和 S_i;
7                  p 自增 1;
8              end
9          end
10         交换 S_0 和 S_p;
11         QuickSort(S, p, l + 1);
12         QuickSort(S' = {S_i | i ∈ [p + 1, n − 1]}, n − (p + 1), l + 1);
13     end
14     return
15 end
```

1.3 算法伪代码

见算法 1。

2 找第 2 小数

2.1 问题描述

Description

给定一维 `int` 型数组，请找到第 2 大的数。

Input

输入第 1 行有一个 `int` 型正整数 `m`(`m`<100)，表示有 m 行输入。

每行输入的第一个数为 int 型正整数 `n`(0<`n`<1000)，后面接着输入 `n` 个 `int` 型整数。

Output

输出 m 行，每行为找第 2 大数。

Sample Input

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

Sample Output

```
2
3
```

2.2 算法思路

2.3 算法伪代码

3 寻找两个正序数组的中位数

3.1 问题描述

Description

给定两个大小为 m 和 n 的正序（从小到大）数组 nums1 和 nums2。

请你找出并返回这两个正序数组的中位数。

进阶：你能设计一个时间复杂度为 $O(log(m+n))$ 的算法解决此问题吗？

Input

第一行输入 nums 表示有 nums 组测试

每组测试输入 n 和 m，分别表示数组 nums1 和 nums2 的长度

然后输入正序数组 nums1

接着输入正序数组 nums2

Output

对每组测试数据输出两个正序数组的中位数

Sample Input

```
2
2 1
1 3
2
2 2
1 2
3 4
```

Sample Output

```
2.00000
2.50000
```

3.2 算法思路

3.3 算法伪代码

4 搜索二维矩阵

4.1 问题描述

Description

编写一个高效的算法来搜索 `m x n` 矩阵 `matrix` 中的一个目标值 `target`。该矩阵具有以下特性：

1. 每行的元素从左到右升序排列。
2. 每列的元素从上到下升序排列。

Input

第一行输入 `nums` 表示有 `nums` 组测试。
每组测试输入 `m`、`n`，`target`，分别表示矩阵的行列数以及目标值。
接下来输入 `m * n` 的二维矩阵。

Output

对每组测试数据输出能否在矩阵中找到 target。
若能找到，输出 `true`。

若找不到，输出 `false`。

Sample Input

```
1
5 5 5
1 4 7 11 15
2 5 8 12 19
3 6 9 16 22
10 13 14 17 24
18 21 23 26 30
```

Sample Output

```
true
```

提示

`m` <= 1000
`n` <= 1000

4.2 算法思路

4.3 算法伪代码