算法设计与分析

关于两个有序数组的中位数寻找

Lecturer: 杨启哲　　Last modified: 2024 年 10 月 27 日

这是一个对 Leetcode 上的题目4. 寻找两个正序数组的中位数的额外解释。本题的描述如下:

问题描述：寻找两个正序数组的中位数

给定两个大小分别为 m 和 n 的正序（从小到大）数组 A 和 B。请找出并返回这两个正序数组的中位数。

我们不再阐述 $O(\log(m+n))$ 的二分查找算法。事实上，我们能给出一个时间复杂度为$O(\log(\min(m,n)))$的算法，算法的基本思想如下:

$$\text{寻找到最大的 } i\text{，使得 } A[i] \leqslant B[\lfloor \frac{m+n+1}{2} \rfloor - i + 1]$$

我们断言，中位数一定在 $A[i+1], A[i], B[\lfloor \frac{m+n+1}{2} \rfloor - i + 1], B[\lfloor \frac{m+n+1}{2} \rfloor - i]$ 这四个数之间产生。从而我们可以通过在较小的那个数组中进行二分查找来找到这个 i，因此算法是$O(\log(\min(m,n)))$的。

下面我们来解释为什么要选择这么一个 i。由对称性，不妨令 $m \leqslant n$. 为了方便讨论，我们对每个数组添加两个虚拟元素，即 $A[0] = B[0] = -\infty, A[m+1] = B[n+1] = \infty$。注意这些元素只是更好的处理边界情况，并不计算在中位数的排序在内。

对任意的 $i \in [0, 1, \dots, m]$ 和 $j \in [0, 1, \dots, n]$，该下标对给出了两个数组的一个划分:

- $V_1 = \{A[1], \dots, A[i], B[1], \dots, B[j]\}$，$|V_1| = i + j$。
- $V_2 = \{A[i+1], \dots, A[m], B[j+1], \dots, B[n]\}$，$|V_2| = m + n - i - j$。

特别的，我们可以令 $j = \lfloor \frac{m+n+1}{2} \rfloor - i$(若 $j > m$ 的话我们忽略该划分)，则我们可以保证:

- 当 $m+n$ 为偶数时，$|V_1| = |V_2|$。
- 当 $m+n$ 为奇数时，$|V_1| = |V_2| + 1$。

选取满足上述条件的 (i,j) 时，两个集合基本上被划分为两个大小相等的部分。此时，如果有:

$$\max_{x \in V_1} x \leqslant \min_{y \in V_2} y$$

也就是说，V_1 中的最大值小于等于 V_2 中的最小值，那么我们就找到了中位数，其为 $\max_{x \in V_1} x$ 或者 $\frac{1}{2}(\max_{x \in V_1} x + \min_{y \in V_2} y)$，并且我们有:

- $\max_{x \in V_1} x = \max\{A[i], B[j]\}$.
- $\min_{y \in V_2} y = \min\{A[i+1], B[j+1]\}$.

因此上述条件转化成了寻找 $i \in [0, 1, \ldots, m]$ 满足:

$$A[i] \leqslant B[j+1] \quad 且 \quad B[j] \leqslant A[i+1]$$

我们进一步证明，这等价于下面的叙述:

寻找到最大的i，使得$A[i] \leqslant B[j+1]$

我们分两步说明:

1. 最大使得 $A[i] \leqslant B[j+1]$ 的 i，一定满足 $A[i] \leqslant B[j+1]$ 且 $B[j] \leqslant A[i+1]$。

   这是因为由定义：$A[i+1] > B[j]$.(否则 i 不是最大的)。

2. 这样的 i 一定存在。

   考虑 A 和 B 两个数组中第 $\lfloor \frac{m+n+1}{2} \rfloor$ 大的数，其可能是 $A[i_0]$ 或者 $B[j_0]$:

   - 如果是 $A[i_0]$，则取 $i = i_0$，并且由 $A[i_0]$ 的大小关系 (两个数组中第 $\lfloor \frac{m+n+1}{2} \rfloor$ 大的数)，会存在 B 中 $\lfloor \frac{m+n+1}{2} \rfloor - i_0$ 个数比起小，从而我们有:

   $$\begin{aligned} A[i_0] \leqslant & B[\lfloor \frac{m+n+1}{2} \rfloor - i_0 + 1] \\ B[\lfloor \frac{m+n+1}{2} \rfloor - i_0] \leqslant & A[i_0] \leqslant A[i_0+1] \end{aligned}$$

   - 如果是 $B[j_0]$，则取 $i = \lfloor \frac{m+n+1}{2} \rfloor - j_0$，类似上面的讨论，我们有:

   $$\begin{aligned} A[\lfloor \frac{m+n+1}{2} \rfloor - j_0] \leqslant & B[j_0] \leqslant B[j_0+1] \\ B[j_0] \leqslant & A[\lfloor \frac{m+n+1}{2} \rfloor - j_0 + 1] \end{aligned}$$

这样便完成了我们全部的证明。

最后我们再概览一下算法的流程, 令 $m \leqslant n$，并且令 A 为较短的数组:

1. 在 A 中二分寻找最大的 i，使得 $A[i] \leqslant B[\lfloor \frac{m+n+1}{2} \rfloor - i + 1]$。
2. A, B 两个数组中第 $\lfloor \frac{m+n+1}{2} \rfloor$ 大的数 m_1 和第 $\lfloor \frac{m+n+1}{2} \rfloor + 1$ 大的数 m_2 满足:

   $$\begin{aligned} m_1 &= \max\{A[i], B[\lfloor \frac{m+n+1}{2} \rfloor - i]\} \\ m_2 &= \min\{A[i+1], B[\lfloor \frac{m+n+1}{2} \rfloor - i + 1]\} \end{aligned}$$

3. 若 $m+n$ 为奇数，则 m_1 即为中位数，否则中位数为 $\frac{1}{2}(m_1 + m_2)$。

Remark 0.1

本文解释的时候为了方便理解，用的数组下标是从 1 开始计算的，因此会与实际编程中的小标会存在一些误差。