

4. Median of Two Sorted Arrays

题目描述: <https://leetcode.com/problems/median-of-two-sorted-arrays/>

给定两个分别排序好的数组，求这两个数组合并在一起的中位数。例如：

```
Example 1:
nums1 = [1, 3]
nums2 = [2]
The median is 2.0

Example 2:
nums1 = [1, 2]
nums2 = [3, 4]
The median is (2 + 3)/2 = 2.5
```

解题思路：

中位数的意思就是按照这个数，可以将本数组分成大小相同的两份，且第一份的元素都比第二份中小。理解了中位数的意思，则本题目求出的结果可以把两个数组 $\text{lenA} \leq \text{lenB}$ 分成

$\text{left} = A[0, i-1] + B[0, j-1]$

$\text{right} = A[i, m-1] + B[j, n-1]$ 两份，使得

$\text{len}(\text{left}) == \text{len}(\text{right})(+1) \ \&\& \ \text{max}(\text{left}) \leq \text{min}(\text{right})$

$\text{len}(\text{left}) == i+j$

$\text{len}(\text{right}) == m-i+n-j$

$i+j == m+n-i-j(+1)$

$i = 0 \sim m$

$j = (m+n+1)/2-i$

因此在 $0 \sim m$ 范围内二分 i 就可以了， i 需要满足的条件为： $A(i-1) \leq B(j) \ \&\& \ A(i) \geq B(j-1)$

如果 $A(i-1) > B(j)$ 则 $r = i-1$;

如果 $A(i) < B(j-1)$ 则 $l = i+1$; 如果 $i == 0 \ \parallel \ j == n$ 则只判断 $A(i) > B(j-1)$

如果 $j == 0 \ \parallel \ i == m$ 则只判断 $A(i-1) < B(j)$

如果 $m+n$ 是偶数，则取 $\text{max}(\text{left}) + \text{min}(\text{right})$

如果 $m+n$ 是奇数，则取 $\text{max}(\text{left})$

代码：

```

class Solution {
public:
    double findMedianSortedArrays(vector<int>& nums1, vector<int>& nums2) {
        if(nums1.size() > nums2.size()) return findMedianSortedArrays(nums2, nums1);

        int m = nums1.size(), n = nums2.size();
        int imax = m, imin = 0;
        int i, j;
        while(imax >= imin) {
            i = (imax + imin) / 2;
            j = (n + m + 1) / 2 - i;
            cout << i << " " << j << endl;
            if((j > 0 && i < m) && nums2[j-1] > nums1[i]) {
                imin = i+1;
            }
            else if((i > 0 && j < n) && nums1[i-1] > nums2[j]) {
                imax = i;
            }
            else {
                break;
            }
        }
        // cout << i << " " << j << endl;

        double maxleft = (i == 0 ? nums2[j-1] : j == 0 ? nums1[i-1] : max(nums1[i-1], nums2[j-1]));
        double minright = (i == m ? nums2[j] : j == n ? nums1[i] : min(nums1[i], nums2[j]));
        // cout << maxleft << " " << minright << endl;
        if((m + n) % 2 == 1) {
            return maxleft;
        }
        else {
            return (maxleft + minright) / 2.0;
        }
    }
};

```