Kerajaan Angka

Author: Albar N

Dahulu kala, ada kerajaan angka, dan dua keluarga paling berkuasa di kerajaan itu adalah *Sorted Arrays* dan *Median Seekers*. *Sorted Arrays* dikenal karena cara mereka yang terorganisir dan efisien, sedangkan *Median Seekers* dikenal karena mereka kemampuan untuk menemukan keseimbangan dan keadilan dalam segala situasi.

Suatu hari, kedua keluarga dipertemukan untuk memecahkan masalah yang telah meresahkan kerajaan. Ada banyak sekali daftar angka, dan orang-orang perlu menemukan median dari daftar gabungan tersebut. *Sorted Arrays* ditugaskan untuk mengatur daftar tersebut, sedangkan *Median Seekers* bertugas menemukan median.

Array Terurut bekerja tanpa lelah untuk mengurutkan daftar, dan setelah selesai, mereka mempresentasikannya kepada *Median Seekers*. *Median Seekers* kemudian mulai bekerja, menggunakan pendekatan *divide and conquer* mereka untuk menemukan median dari daftar gabungan.

Bantulah keluarga *Median Seekers* untuk mencari nilai median dari kedua array a dan array b yang diberikan.

Input Format

- Baris pertama berisi array a.
- Baris kedua berisi array b.

Constraints

- Array a dan b sudah dipastikan terurut berkat perjuangan keluarga *Sorted Arrays*.
- Elemen array a dan b bilangan bulat.

Output Format

Median dari kedua array jika digabungkan, dengan satu angka dibelakang koma.

Sample Input 0

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

Sample Output 0

```
4.0
```

Sample Input 1

```
1 3
2
```

Sample Output 1

```
2.0
```

Contest ends in 2 days

Submissions: 99
Max Score: 100
Difficulty: Medium

Rate This Challenge:

More

Python 3

```python
def findMedianSortedArrays(nums1, nums2):
    m, n = len(nums1), len(nums2)
    if m > n:
        nums1, nums2, m, n = nums2, nums1, n, m
    imin, imax, half_len = 0, m, (m + n + 1) // 2
    while imin <= imax:
        i = (imin + imax) // 2
        j = half_len - i
        if i < m and nums2[j-1] > nums1[i]:
            imin = i + 1
        elif i > 0 and nums1[i-1] > nums2[j]:
            imax = i - 1
        else:
            if i == 0:
                max_of_left = nums2[j-1]
            elif j == 0:
                max_of_left = nums1[i-1]
            else:
                max_of_left = max(nums1[i-1], nums2[j-1])
            if (m + n) % 2 == 1:
                return float(max_of_left)
            if i == m:
                min_of_right = nums2[j]
            elif j == n:
                min_of_right = nums1[i]
            else:
                min_of_right = min(nums1[i], nums2[j])
            return (max_of_left + min_of_right) / 2.0

nums1 = list(map(int, input().split()))
nums2 = list(map(int, input().split()))
print(findMedianSortedArrays(nums1, nums2))
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code