

[LintCode领扣题解 \(/problem\)](#) / [合并排序数组 II · Merge Two Sorted Arrays](#)

合并排序数组 II · Merge Two Sorted Arrays

中文

[LintCode 版权所有 \(/problem/?tags=lintcode-copyright\)](#)[排序数组 \(/problem/?tags=sorted-array\)](#)[数组 \(/problem/?tags=array\)](#)

描述

合并两个有序升序的整数数组A和B变成一个新的数组。新数组也要有序。

样例

样例 1:

输入: A=[1], B=[1]
输出: [1,1]
样例解释: 返回合并后的数组。

样例 2:

输入: A=[1,2,3,4], B=[2,4,5,6]
输出: [1,2,2,3,4,4,5,6]
样例解释: 返回合并后的数组。

挑战

你能否优化你的算法, 如果其中一个数组很大而另一个数组很小?

在线评测地址: <https://www.lintcode.com/problem/merge-two-sorted-arrays/> (<https://www.lintcode.com/problem/merge-two-sorted-arrays/>)

[收起题目描述 ^](#)

语言类型

[ALL \(20\)](#)[python \(7\)](#)[java \(5\)](#)[cpp \(3\)](#)[javascript \(3\)](#)[golang \(2\)](#)[上传题解](#)**令狐冲**

更新于 8/19/2020, 1:12:04 PM

使用两个指针分别对数组从小到大遍历, 每次取二者中较小的放在新数组中。直到某个指针先到结尾, 另一个数组中剩余的数字直接放在新数组后面。

时间复杂度O(n)

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution {
    /**
     * @param A and B: sorted integer array A and B.
     * @return: A new sorted integer array
     */
    public int[] mergeSortedArray(int[] A, int[] B) {
        if (A == null || B == null) {
            return null;
        }

        int[] result = new int[A.length + B.length];
        int i = 0, j = 0, index = 0;

        while (i < A.length && j < B.length) {
            if (A[i] < B[j]) {
                result[index++] = A[i++];
            } else {
                result[index++] = B[j++];
            }
        }

        while (i < A.length) {
            result[index++] = A[i++];
        }
        while (j < B.length) {
            result[index++] = B[j++];
        }

        return result;
    }
}
```

👍 获赞 6

💬 2 条评论

你的口袋题库

2000+ 算法真题、国内外名企题库免费开放



九章算法APP

华助教

更新于 6/9/2020, 7:04:27 AM

使用两个指针分别对数组从小到大遍历, 每次取二者中较小的放在新数组中。直到某个指针先到结尾, 另一个数组中剩余的数字直接放在新数组后面。

时间复杂度 $O(n)$

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
const mergeSortedArray = function (A, B) {
    var ret = [];
    var i, j
    for (len = 0, i = 0, j = 0; i < A.length && j < B.length; ) {
        if (A[i] < B[j]) {
            ret.push(A[i++]);
        } else {
            ret.push(B[j++]);
        }
    }
    while (i < A.length) {
        ret.push(A[i++]);
    }
    while (j < B.length) {
        ret.push(B[j++]);
    }
    return ret;
}
```

 获赞 0 添加评论

令狐冲

更新于 6/9/2020, 7:04:20 AM

使用两个指针分别对数组从小到大遍历, 每次取二者中较小的放在新数组中。直到某个指针先到结尾, 另一个数组中剩余的数字直接放在新数组后面。

时间复杂度 $O(n)$

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ /
Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution:
    """
    @param A: sorted integer array A
    @param B: sorted integer array B
    @return: A new sorted integer array
    """
    def mergeSortedArray(self, A, B):
        i, j = 0, 0
        C = []
        while i < len(A) and j < len(B):
            if A[i] < B[j]:
                C.append(A[i])
                i += 1
            else:
                C.append(B[j])
                j += 1
        while i < len(A):
            C.append(A[i])
            i += 1
        while j < len(B):
            C.append(B[j])
            j += 1

        return C
```

👍 获赞 0

💬 1 条评论



九章用户N2FZTL

更新于 6/9/2020, 7:03:48 AM

来一个简单易读的follow up解法, binary search

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
public class Solution {
    /**
     * @param A: sorted integer array A
     * @param B: sorted integer array B
     * @return: A new sorted integer array
     */
    public int[] mergeSortedArray(int[] A, int[] B) {
        if(A == null || A.length == 0) return B;
        if(B == null || B.length == 0) return A;
        int[] res = new int[A.length + B.length];
        int idx = 0;
        int pa = 0;
        for(int i=0;i<B.length;i++){
            int position = binarySearch(A, B[i]);
            while(pa < position){
                res[idx++] = A[pa++];
            }
            res[idx++] = B[i];
        }
        while(pa < A.length){
            res[idx++] = A[pa++];
        }
        return res;
    }

    private int binarySearch(int[] A, int target){
        int left = 0;
        int right = A.length-1;
        while(left <= right){
            int mid = left + (right-left)/2;
            if(A[mid] == target) return mid;
            if(target < A[mid]){
                right = mid - 1;
            }else{
                left = mid + 1;
            }
        }
        return left;
    }
}
```

👍 获赞 3

💬 1 条评论

**ctc316**

更新于 6/9/2020, 7:03:50 AM

Python 版本的 Binary search and insert Time: $O(m \log n)$, should be faster if $n \gg m$

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    """
    @param A: sorted integer array A
    @param B: sorted integer array B
    @return: A new sorted integer array
    """
    def mergeSortedArray(self, A, B):
        # make B the smaller one
        if len(A) < len(B):
            temp = A
            A = B
            B = temp

        # binary search and insert
        for ele in B:
            left = 0
            right = len(A) - 1
            while left + 1 < right:
                mid = int(left + (right - left) / 2)
                if ele < A[mid]:
                    right = mid
                else:
                    left = mid

            if ele < A[left]:
                A.insert(left, ele)
            elif ele > A[right]:
                A.insert(right + 1, ele)
            else:
                A.insert(right, ele)

        return A

```

👍 获赞 2

💬 5 条评论



九章用户Y1KOB5

更新于 6/9/2020, 7:03:57 AM

I'm using binary search for the experience case when A is very big while B is very small Binary search B's element in A

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
public class Solution {
    /**
     * @param A: sorted integer array A
     * @param B: sorted integer array B
     * @return: A new sorted integer array
     */

```

```
*/
public int[] mergeSortedArray(int[] A, int[] B) {
    if (A == null || A.length == 0) {
        return B;
    }

    if (B == null || B.length == 0) {
        return A;
    }

    int[] result = new int[A.length + B.length];
    int[] target;
    int[] source;
    if (A.length > B.length) {
        target = A;
        source = B;
    } else {
        target = B;
        source = A;
    }

    Map<Integer, List<Integer>> positionMap = new HashMap<>();

    for (int i = 0; i < source.length; i++) {
        int num = source[i];
        int l = 0;
        int r = target.length - 1;
        while (l + 1 < r) {
            int mid = (r - l) / 2 + l;
            if (target[mid] > num) {
                r = mid;
            } else {
                l = mid;
            }
        }
        if (target[l] > num) {
            if (!positionMap.containsKey(l - 1)) {
                positionMap.put(l - 1, new ArrayList<>());
            }
            positionMap.get(l - 1).add(num);
        } else if (target[r] > num) {
            if (!positionMap.containsKey(l)) {
                positionMap.put(l, new ArrayList<>());
            }
            positionMap.get(l).add(num);
        } else {
            if (!positionMap.containsKey(r)) {
                positionMap.put(r, new ArrayList<>());
            }
            positionMap.get(r).add(num);
        }
    }

    int resultC = 0;
    for (int i = -1; i < target.length; i++) {
        if (i >= 0) {
            result[resultC] = target[i];
            resultC++;
        }
        if (positionMap.get(i) != null && positionMap.get(i).size() > 0) {
            for (int j = 0; j < positionMap.get(i).size(); j++) {
                result[resultC] = positionMap.get(i).get(j);
                resultC++;
            }
            positionMap.remove(i);
        }
    }
}
```

```
        return result;
    }
}
```

👍 获赞 1 💬 1 条评论



abtion

更新于 6/9/2020, 7:03:57 AM

主要考察了循环和Vector的使用。很简单的O(n)级算法, 注意考虑各种情况就好。

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution {
public:
    /**
     * @param A: sorted integer array A
     * @param B: sorted integer array B
     * @return: A new sorted integer array
     */
    vector<int> mergeSortedArray(vector<int> &A, vector<int> &B) {
        // write your code here
        vector<int> rst(1,A[0]);
        int i = 0,j=0;
        while(i<A.size()&&j<B.size()){
            if(A[i]<B[j]){
                rst.insert(rst.end(),A[i++]);
            }else{
                rst.insert(rst.end(),B[j++]);
            }
        }
        if(i<A.size()){
            rst.insert(rst.end(),A.begin()+i,A.end());
        }else if(j<B.size()){
            rst.insert(rst.end(),B.begin()+j,B.end());
        }
        rst.erase(rst.begin());
        return rst;
    }
};
```

👍 获赞 1 💬 添加评论



Noise

更新于 6/9/2020, 7:03:57 AM

For the follow up challenge: one is large and the other is small


```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution:
    """
    @param A: sorted integer array A
    @param B: sorted integer array B
    @return: A new sorted integer array
    """
    def mergeSortedArray(self, A, B):
        # find the last insertion position
        from bisect import bisect_right

        small_arr = large_arr = None
        if len(A) < len(B):
            small_arr, large_arr = A, B
        else:
            small_arr, large_arr = B, A

        i = 0
        result = []
        for num in small_arr:
            pos = bisect_right(large_arr, num)
            result += large_arr[i: pos]
            result.append(num)
            i = pos
        result += large_arr[i:]

        return result
```

👍 获赞 1 💬 添加评论



九章用户8ZBNY6

更新于 6/9/2020, 7:04:27 AM

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    def mergeSortedArray(self, A, B):
        # write your code here
        result = []
        tt = []
        for i in A:
            for j in B:
                if i >= j:
                    # 避免内层循环重复判断
                    result.append(j)
                    B = B[1:]
                else:
                    # 因列表有序, 此时 i 小于 j 及其后续所有元素, 故可直接结束此轮循环
                    result.append(i)
                    A = A[1:]
                    break
            if A == []:
                tt = B
            if B == []:
                tt = A
        return result + tt

# 验证
A1 = [3, 7, 8, 9, 12]
B1 = [5, 6, 10, 13, 25, 30]
A2 = [1, 2, 3, 4]
B2 = [2, 4, 5, 6]
A3 = [7]
B3 = [5, 7]
As = [A1, A2, A3]
Bs = [B1, B2, B3]

res = Solution()
for A in As:
    for B in Bs:
        print(res.mergeSortedArray(A, B))
```

👍 获赞 0 💬 添加评论



九章用户WP7TG8

更新于 6/9/2020, 7:04:26 AM

lintcode时间1087 ms 思路: 先两两比较, 把小的加到c里面, 然后自加。剩余的最后加到c

课程 (/course/)

旗舰课 (/premium-course/)

1对1私教 (/1on1/)

免费课

Seminar

邀请有礼

视频题解

成功案

消息

更多...

我的课程 (/accounts/profile/)

APP

邀请有礼

视频题解

成功案

消息

更多...

我的课程 (/accounts/profile/)

```
/**
 * 本参考程序由九章算法用户提供。版权所有，转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作，授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括：九章算法班 2020升级版，算法强化班，算法基础班，北美算法面试高频题班，Java 高级工程师 P6+ 小班课，面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括：系统设计 System Design，面向对象设计 OOD
 * - 专题及项目类课程包括：动态规划专题班，Big Data - Spark 项目实战，Django 开发项目课
 * - 更多详情请见官方网站：http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    def mergeSortedArray(self, A, B):
        p1=len(A)
        p2=len(B)
        i,j=0,0
        c=[]
        while (i<p1) and (j<p2):
            if A[i]<B[j]:
                c.append(A[i])
                i+=1
            else:
                c.append(B[j])
                j+=1
        while i < p1:
            c.append(A[i])
            i += 1
        while j < p2:
            c.append(B[j])
            j += 1
        return c
```

👍 获赞 0 💬 添加评论

九章用户I35N06

更新于 6/9/2020, 7:04:26 AM

I use call system function to sort and merge



invitation/sh:



```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
public int[] mergeSortedArray(int[] A, int[] B) {
    // write your code here
    int [] result = new int[A.length + B.length];
    int count = 0;
    for(int i = 0; i < A.length; i++) {
        result[i] = A[i];
        count++;
    }
    for(int j = 0 ; j < B.length; j++) {
        result[j+count] = B[j];
    }

    Arrays.sort(result);
    for(int i = 0; i < result.length; i++) {
        System.out.println(result[i]);
    }
    return result;
}
}

```

👍 获赞 0

💬 添加评论

**九章用户W5APIL**

更新于 6/9/2020, 7:04:25 AM

reference: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/push (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/push)

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
/**
 * @param A: sorted integer array A
 * @param B: sorted integer array B
 * @return: A new sorted integer array
 */
// let sum1 = [1,3,5,8,10,6];
// let sum2 = [2,6,3,9];
const mergeSortedArray = function (A, B) {
    Array.prototype.push.apply(A,B);
    A.sort(function(a,b){
        return a - b;});
    return A;
};

```

👍 获赞 0

💬 添加评论

**julyly**

更新于 6/9/2020, 7:04:25 AM

不是很懂vector,按照数组的方式写的。所有测试点运行总时间285ms

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution {
public:
    /**
     * @param A: sorted integer array A
     * @param B: sorted integer array B
     * @return: A new sorted integer array
     */
    vector<int> mergeSortedArray(vector<int> &A, vector<int> &B) {
        int i=0,j=0;
        vector<int>C;
        for(i=0,j=0;i<A.size()&&j<B.size();){
            if(A[i]<B[j])
            {
                C.push_back(A[i]);
                i++;
            }
            else if(A[i]>B[j])
            {
                C.push_back(B[j]);
                j++;
            }
            else if(A[i]==B[j])
            {
                C.push_back(A[i]);
                C.push_back(B[j]);
                i++;
                j++;
            }
        }
        for(int m=i;m<A.size();m++)
        {
            C.push_back(A[m]);
        }
        for(int m=j;m<B.size();m++)
        {
            C.push_back(B[m]);
        }
        return C;
    }
};
```

👍 获赞 0

💬 添加评论

**九章用户6XWOIY**

更新于 6/9/2020, 7:04:24 AM

same as Merge Function in MergeSort algorithms

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution {
public:
    vector<int> mergeSortedArray(vector<int> &A, vector<int> &B) {
        vector<int> c;
        int i=0,j=0;
        while(i<A.size() && j<B.size())
            if(A[i]<B[j])
                c.push_back(A[i++]);
            else
                c.push_back(B[j++]);
        while(i<A.size())
            c.push_back(A[i++]);
        while(j<B.size())
            c.push_back(B[j++]);
        return c;
    }
};

```

👍 获赞 0

💬 添加评论

**九章用户PDJKTI**

更新于 6/9/2020, 7:04:23 AM

新数组,用函数sort() 排序

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    """
    @param A: sorted integer array A
    @param B: sorted integer array B
    @return: A new sorted integer array
    """
    def mergeSortedArray(self, A, B):
        # write your code here
        C=A+B
        C.sort(reverse = False)
        return C

```

👍 获赞 0

💬 2 条评论

**九章用户G1LOUJ**

更新于 6/9/2020, 7:04:17 AM

Merge Two Sorted Arrays, 使用一个for循环实现, 双指针方法。

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ /
Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    """
    @param A: sorted integer array A
    @param B: sorted integer array B
    @return: A new sorted integer array
    """
    def mergeSortedArray(self, A, B):
        if not A:
            return B
        if not B:
            return A

        len_a, len_b = len(A), len(B)
        n = len_a + len_b

        res = [0 for _ in range(n)]
        i, j = 0, 0

        for k in range(n):
            if i < len_a and (j >= len_b or A[i] <= B[j]):
                res[k] = A[i]
                i += 1
            else:
                res[k] = B[j]
                j += 1

        return res
```

👍 获赞 0

💬 添加评论



bender

更新于 6/9/2020, 7:04:12 AM

已知两个数组都为有序, 那么可以按顺序比较相关的元素, 最后可以利用golang简介的数组拷贝在其中一个数组遍历结束后停止对之后的元素的判断

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
func mergeSortedArray (A []int, B []int) []int {
    return combine(A,B)
}

func combine(baseArr []int, member []int)( all []int) {
    i := 0
    j := 0
    for {
        if j == len(member) {
            all = append(all, baseArr[i:]...)
            break
        }
        if i == len(baseArr) {
            all = append(all, member[j:]...)
            break
        }

        if baseArr[i] > member[j] {
            all = append(all, member[j])
            j++
        } else if baseArr[i] < member[j] {
            all = append(all, baseArr[i])
            i++
        } else {
            all = append(all, baseArr[i], member[j])
            i++
            j++
        }
    }
    return
}
```

👍 获赞 0

💬 添加评论

**Eddy**

更新于 6/9/2020, 7:04:08 AM

Traverse two arrays and add elements to a new array by order, until one of the arrays is exhausted. Then add all elements in the other array to the new array.


```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
public class Solution {
    /**
     * @param A: sorted integer array A
     * @param B: sorted integer array B
     * @return: A new sorted integer array
     */
    public int[] mergeSortedArray(int[] A, int[] B) {
        // write your code here
        if (A == null) {
            return B;
        }
        if (B == null) {
            return A;
        }
        int[] result = new int[A.length + B.length];
        int i = 0, j = 0, k = 0;
        while (i < A.length && j < B.length) {
            result[k++] = A[i] < B[j] ? A[i++] : B[j++];
        }

        while (i < A.length) {
            result[k++] = A[i++];
        }

        while (j < B.length) {
            result[k++] = B[j++];
        }

        return result;
    }
}
```

👍 获赞 0 💬 添加评论



九章用户ELUPCW

更新于 6/9/2020, 7:04:08 AM

合并两个有序升序的整数数组A和B变成一个新的数组。新数组也要有序。

主要依靠 二分查找, 并且利用 有序、升序, 可以在下轮查找时, 可以缩小被查找的分片的范围:

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
/**
 * @param A: sorted integer array A
 * @param B: sorted integer array B
 * @return: A new sorted integer array
 */
func mergeSortedArray (A []int, B []int) []int {
    // write your code here
    a := []int{}
    b := []int{}
    for _, n := range A {
        a, B = binarySearchIndex(B, n)
        b = append(b, a...)
    }
    return append(b, B...)
}

func binarySearchIndex(L []int, n int) ([]int, []int) {
    l := len(L)
    if l == 0 {
        return []int{n}, []int{}
    } else if l == 2 {
        if n > L[1] {
            return []int{L[0], L[1], n}, []int{}
        } else if n < L[0] {
            return []int{n}, L
        } else {
            return []int{L[0], n}, []int{L[1]}
        }
    } else if l == 1 {
        if n > L[0] {
            return L, []int{n}
        } else {
            return []int{n}, L
        }
    } else {
        if L[l/2] == n {
            left := make([]int, l/2)
            copy(left, L[:l/2])
            right := make([]int, l-l/2)
            copy(right, L[l/2:])

            return append(left, n), right
        }
        if L[l/2] > n {
            left := make([]int, l/2+1)
            copy(left, L[:l/2+1])
            right := make([]int, l-l/2-1)
            copy(right, L[l/2+1:])
            a, b := binarySearchIndex(left, n)
            return a, append(b, right...)
        } else {
            a, b := binarySearchIndex(L[l/2:], n)
            return append(L[:l/2], a...), b
        }
    }
}

```

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
func mergeSortedArray (A []int, B []int) []int {
    // write your code here
    a := []int{}
    b := []int{}
    for _, n := range A {
        a, B = binarySearchIndex(B, n)
        b = append(b, a...)
    }
    return append(b, B...)
}

func binarySearchIndex(L []int, n int) ([]int, []int) {
    l := len(L)
    if l == 0 {
        return []int{n}, []int{}
    } else if l == 2 {
        if n > L[1] {
            return []int{L[0], L[1], n}, []int{}
        } else if n < L[0] {
            return []int{n}, L
        } else {
            return []int{L[0], n}, []int{L[1]}
        }
    } else if l == 1 {
        if n > L[0] {
            return L, []int{n}
        } else {
            return []int{n}, L
        }
    } else {
        if L[l/2] == n {
            left := make([]int, l/2)
            copy(left, L[:l/2])
            right := make([]int, l-l/2)
            copy(right, L[l/2:])

            return append(left, n), right
        }
        if L[l/2] > n {
            left := make([]int, l/2+1)
            copy(left, L[:l/2+1])
            right := make([]int, l-l/2-1)
            copy(right, L[l/2+1:])
            a, b := binarySearchIndex(left, n)
            return a, append(b, right...)
        } else {
            a, b := binarySearchIndex(L[l/2:], n)
            return append(L[:l/2], a...), b
        }
    }
}

```

👍 获赞 0 💬 添加评论

**king-xiyu**

更新于 6/9/2020, 7:04:05 AM

分享俩个运行时间短的写法; 用到的方法: concat方法用于连接两个或多个数组 sort方法用于对数组的元素进行排序 es6新特性 ...扩展符 其中要注意sort方法的使用, 推荐参考: <http://c.biancheng.net/view/5669.html> (<http://c.biancheng.net/view/5669.html>)

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
/**
 * @param A: sorted integer array A
 * @param B: sorted integer array B
 * @return: A new sorted integer array
 */

// 方法一
const mergeSortedArray = function (A, B) {
  // concat方法用于连接两个或多个数组
  let array = A.concat(B)
  // sort方法用于对数组的元素进行排序
  return array.sort((a, b) => {return a - b})
}

// 方法二
const mergeSortedArray = function (A, B) {
  // es6新特性 把俩个数组里的参数放到新的数组里
  let array = [...A, ...B]
  return array.sort((a, b) => {return a - b})
}
```

👍 获赞 0

💬 添加评论

进阶课程

视频+互动

直播+互动

直播+互动

互动课

九章算法班 2021 版

8周时间精通 57 个核心高频考点, 9 招击破 FLAG、BATJ 算法面试。22....

系统架构设计 System Design 2021 版

成为百万架构师必上。30 课时带你快速掌握18大系统架构设计知识点与面...

九章算法面试高频题冲刺班

每期更新 15% 题目, 考前押题, 一举拿下FLAG & BATJ Offer

面向对象设计 OOD

应届生及亚马逊面试必考, IT求职必备基础

[首页 \(/?skip_redirect=true\)](/?skip_redirect=true)


[联系我们 \(mailto:info@jiuzhang.com\)](mailto:info@jiuzhang.com)


[加入我们 \(/joinus\)](/joinus)

Copyright © 2013-2020 九章算法 浙ICP备19045946号-1

(<http://www.miibeian.gov.cn/>)

商务合作: fukesu@jiuzhang.com (<mailto:fukesu@jiuzhang.com>)

 (<http://weibo.com/ninechapter>)

 (<https://www.zhihu.com/people/crackinterview/>)

(/)