



Aror University of Art, Architecture, Design & Heritage Sukkur

Department of AI-Multimedia and Gaming

Lab 6: Sorting: Bubble Sort

Date:01 Oct, 2024

Subject: Data Structure (CSC221), Fall 2024

Instructor: Abdul Ghafoor

Lab objectives: The objective of this lab is to understand and implement the Bubble Sort algorithm.

- [Leet Code](#)

<https://leetcode.com/problems/merge-sorted-array/description/?envType=problem-list-v2&envId=sorting>

Submit Ctrl Enter Accepted X

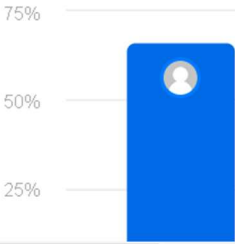
All Submissions

Accepted
submitted at Oct 1

Editor

Runtime
0 ms | Beats 100.00% 🏆
[Analyze Complexity](#)

Memory
42.26 MB | Beats 29.28%



</> Code

Java Auto

```
1 class Solution {
2     public void merge(int[] nums1, int m, int[]
  nums2, int n) {
3         int index1 = m - 1;
4         int index2 = n - 1;
5         int mergeIndex = m + n - 1;
6
7         while (index1 >= 0 && index2 >= 0) {
8             if (nums1[index1] > nums2[index2]) {
9                 nums1[mergeIndex--] = nums1
[index1--];
10            } else {
11                nums1[mergeIndex--] = nums2
[index2--];
12            }
13        }
14        while (index2 >= 0) {
```

Saved Ln 13, Col 10

☒ Testcase > Test Result

Output

[1,2,2,3,5,6]

<https://leetcode.com/problems/intersection-of-two-arrays-ii/description/?envType=problem-list-v2&envId=sorting>

```
1  class Solution {
2      public int[] intersect(int[] nums1, int[] nums2) {
3          int[] count = new int[1001];
4          List<Integer> intersectionList = new ArrayList<>();
5
6
7          for (int num : nums1) {
8              count[num]++;
9          }
10
11
12         for (int num : nums2) {
13             if (count[num] > 0) {
14                 intersectionList.add(num);
15                 count[num]--;
16             }
17         }
18     }
```

Saved

☒ Testcase | [> Test Result](#)

Output

[2, 2]

<https://leetcode.com/problems/find-the-difference/description/?envType=problem-list-v2&envId=sorting>

```
1 class Solution {
2     public int[] intersect(int[] nums1, int[] nums2) {
3         int[] count = new int[1001];
4         List<Integer> intersectionList = new ArrayList<>();
5
6
7         for (int num : nums1) {
8             count[num]++;
9         }
10
11
12         for (int num : nums2) {
13             if (count[num] > 0) {
14                 intersectionList.add(num);
15                 count[num]--;
16             }
17         }
18     }
19 }
```

Saved

☒ Testcase | ☒ Test Result

Output

[2,2]