

# Interview Questions: Mergesort (ungraded)

3/3 points (100%)

Practice Quiz, 3 questions

✓ **Congratulations! You passed!**

Next Item



1 / 1  
points

1.

**Merging with smaller auxiliary array.** Suppose that the subarray  $a[0]$  to  $a[n - 1]$  is sorted and the subarray  $a[n]$  to  $a[2 * n - 1]$  is sorted. How can you merge the two subarrays so that  $a[0]$  to  $a[2 * n - 1]$  is sorted using an auxiliary array of length  $n$  (instead of  $2n$ )?

*Note: these interview questions are ungraded and purely for your own enrichment. To get a hint, submit a solution.*

```
merge a[n] a[0...n-1] to temp[n]
merge a[n] a[0...n-1] to a[0...n]
merge [0...n] temp[n]
```

Your answer cannot be more than 10000 characters.

**Thank you for your response.**

*Hint: copy only the left half into the auxiliary array.*



1 / 1  
points

## Interview Questions: Mergesort (ungraded)

Practice Quiz, 3 questions

3/3 points (100%)

2.

**Counting inversions.** An *inversion* in an array  $a[]$  is a pair of entries  $a[i]$  and  $a[j]$  such that  $i < j$  but  $a[i] > a[j]$ . Given an array, design a linearithmic algorithm to count the number of inversions.

merge

Your answer cannot be more than 10000 characters.



**Thank you for your response.**

*Hint:* count while mergesorting.



1 / 1  
points

3.

**Shuffling a linked list.** Given a singly-linked list containing  $n$  items, rearrange the items uniformly at random. Your algorithm should consume a logarithmic (or constant) amount of extra memory and run in time proportional to  $n \log n$  in the worst case.

merge random two list

Your answer cannot be more than 10000 characters.



**Thank you for your response.**

*Hint:* design a linear-time subroutine that can take two uniformly shuffled linked lists of sizes  $n_1$  and  $n_2$  and combine them into a uniformly shuffled linked list of size  $n_1 + n_2$ .

Interview Questions: Mergesort (ungraded)

Practice Quiz, 3 questions

3/3 points (100%)

