✓ 1 / 1 points

1.

Merging with smaller auxiliary array. Suppose that the subarray $\mathtt{a[0]}$ to $\mathtt{a[n-1]}$ is sorted and the subarray $\mathtt{a[n]}$ to $\mathtt{a[2*n-1]}$ is sorted. How can you merge the two subarrays so that $\mathtt{a[0]}$ to $\mathtt{a[2*n-1]}$ 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.

d

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

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.

d

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.

d

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 combined them into a uniformly shuffled linked lists of size $n_1 + n_2$.