<https://hackernoon.com/50-data-structure-and-algorithms-interview-questions-for-programmers-b4b1ac61f5b0>

# 50+ Data Structure and Algorithms Interview Questions for Programmers

1. **How do you find the missing number in a given integer array of 1 to 100? (**[**solution**](http://javarevisited.blogspot.com/2014/11/how-to-find-missing-number-on-integer-array-java.html?source=post_page---------------------------&ref=hackernoon.com)**)**
2. **How do you find the duplicate number on a given integer array? (**[**solution**](http://javarevisited.blogspot.com/2014/01/how-to-remove-duplicates-from-array-java-without-collection-API.html?source=post_page---------------------------&ref=hackernoon.com)**)**
3. **How do you find the largest and smallest number in an unsorted integer array? (**[**solution**](http://java67.blogspot.com/2014/02/how-to-find-largest-and-smallest-number-array-in-java.html?source=post_page---------------------------&ref=hackernoon.com)**)**
4. **How do you find all pairs of an integer array whose sum is equal to a given number? (**[**solution**](http://javarevisited.blogspot.com/2014/08/how-to-find-all-pairs-in-array-of-integers-whose-sum-equal-given-number-java.html?source=post_page---------------------------&ref=hackernoon.com)**)**
5. **How do you find duplicate numbers in an array if it contains multiple duplicates? (**[**solution**](http://javarevisited.blogspot.com/2014/03/3-ways-to-find-first-non-repeated-character-String-programming-problem.html?source=post_page---------------------------&ref=hackernoon.com)**)**
6. **How are duplicates removed from a given array in Java? (**[**solution**](http://javarevisited.blogspot.com/2014/01/how-to-remove-duplicates-from-array-java-without-collection-API.html?source=post_page---------------------------&ref=hackernoon.com)**)**
7. **How is an integer array sorted in place using the quicksort algorithm? (**[**solution**](http://javarevisited.blogspot.com/2014/08/quicksort-sorting-algorithm-in-java-in-place-example.html?source=post_page---------------------------&ref=hackernoon.com)**)**
8. **How do you remove duplicates from an array in place? (**[**solution**](http://javarevisited.blogspot.com/2014/01/how-to-remove-duplicates-from-array-java-without-collection-API.html?source=post_page---------------------------&ref=hackernoon.com)**)**
9. **How do you reverse an array in place in Java? (**[**solution**](http://javarevisited.blogspot.com/2013/03/how-to-reverse-array-in-java-int-String-array-example.html?source=post_page---------------------------&ref=hackernoon.com)**)**
10. **How are duplicates removed from an array without using any library? (**[**solution**](http://javarevisited.blogspot.sg/2014/01/how-to-remove-duplicates-from-array-java-without-collection-API.html?source=post_page---------------------------&ref=hackernoon.com)**)**

These questions will not only help you to develop your problem-solving skills but also improve your knowledge of array data structure.

If you need more advanced questions based upon array then you can see also see [**The Coding Interview Bootcamp: Algorithms + Data Structures**](https://click.linksynergy.com/fs-bin/click?id=JVFxdTr9V80&subid=0&offerid=323058.1&type=10&tmpid=14538&RD_PARM1=https%3A%2F%2Fwww.udemy.com%2Fcoding-interview-bootcamp-algorithms-and-data-structure%2F&source=post_page---------------------------&ref=hackernoon.com), a bootcamp style course on algorithms, especially designed for  
interview preparation to get a job on technical giants like Google,  
Microsoft, Apple, Facebook etc.

And, if you feel 10 is not enough questions and you need more practice, then you can also check out this list of [**30 array questions**](http://javarevisited.blogspot.sg/2015/06/top-20-array-interview-questions-and-answers.html?source=post_page---------------------------&ref=hackernoon.com).

**2. Linked List Programming Interview Questions**

A [linked list](http://www.java67.com/2017/06/5-difference-between-array-and-linked.html?source=post_page---------------------------&ref=hackernoon.com) is another common data structure that complements the array data structure. Similar to the array, it is also a linear data structure and  
stores elements in a linear fashion.

However, unlike the array, it doesn’t store them in contiguous locations; instead, they are scattered everywhere in memory, which is connected to each other using nodes.

A linked list is nothing but a list of nodes where each node contains the value stored and the address of the next node.

Because of this structure, **it’s easy to add and remove elements in a linked list**, as you just need to change the link instead of creating the array, but the search is difficult and often requires O(n) time to find an element in the singly linked list.

This [article](http://javarevisited.blogspot.sg/2013/07/difference-between-array-and-linked-list-java.html?source=post_page---------------------------&ref=hackernoon.com) provides more information on the difference between an array and linked list data structures.

It also comes in varieties like a singly linked list, which allows you to traverse in one direction (forward or reverse); a **doubly linked list**,  
which allows you to traverse in both directions (forward and backward);  
and finally, the circular linked list, which forms a circle.

In order to solve linked list-based questions, a good knowledge of [recursion](https://javarevisited.blogspot.com/2017/03/how-to-reverse-linked-list-in-java-using-iteration-and-recursion.html?source=post_page---------------------------&ref=hackernoon.com) is important, because **a linked list is a recursive data structure**.

If you take one node from a linked list, the remaining data structure is  
still a linked list, and because of that, many linked list problems have  
simpler recursive solutions than iterative ones.

Here are some of the most common and popular linked list interview questions and their solutions:

1. **How do you find the middle element of a singly linked list in one pass? (**[**solution**](http://javarevisited.blogspot.sg/2012/12/how-to-find-middle-element-of-linked-list-one-pass.html?source=post_page---------------------------&ref=hackernoon.com)**)**
2. **How do you check if a given linked list contains a cycle? How do you find the starting node of the cycle? (**[**solution**](http://javarevisited.blogspot.sg/2013/05/find-if-linked-list-contains-loops-cycle-cyclic-circular-check.html?source=post_page---------------------------&ref=hackernoon.com)**)**
3. **How do you reverse a linked list? (**[**solution**](http://www.java67.com/2016/07/how-to-reverse-singly-linked-list-in-java-example.html?source=post_page---------------------------&ref=hackernoon.com)**)**
4. **How do you reverse a singly linked list without recursion? (**[**solution**](http://javarevisited.blogspot.sg/2017/03/how-to-reverse-linked-list-in-java-using-iteration-and-recursion.html?source=post_page---------------------------&ref=hackernoon.com)**)**
5. **How are duplicate nodes removed in an unsorted linked list? (**[**solution**](https://www.geeksforgeeks.org/remove-duplicates-from-an-unsorted-linked-list/?source=post_page---------------------------&ref=hackernoon.com)**)**
6. **How do you find the length of a singly linked list? (**[**solution**](http://javarevisited.blogspot.sg/2016/05/how-do-you-find-length-of-singly-linked.html?source=post_page---------------------------&ref=hackernoon.com)**)**
7. **How do you find the third node from the end in a singly linked list? (**[**solution**](http://javarevisited.blogspot.sg/2016/07/how-to-find-3rd-element-from-end-in-linked-list-java.html?source=post_page---------------------------&ref=hackernoon.com)**)**
8. **How do you find the sum of two linked lists using Stack? (**[**solution**](https://www.geeksforgeeks.org/sum-of-two-linked-lists/?source=post_page---------------------------&ref=hackernoon.com)**)**

These questions will help you to develop your problem-solving skills as well  
as improve your knowledge of the linked list data structure.

If you are having trouble solving these linked list coding questions then I  
suggest you refresh your data structure and algorithms skill by going  
through [**Data Structures and Algorithms: Deep Dive**](https://click.linksynergy.com/fs-bin/click?id=JVFxdTr9V80&subid=0&offerid=323058.1&type=10&tmpid=14538&RD_PARM1=https%3A%2F%2Fwww.udemy.com%2Fdata-structures-and-algorithms-deep-dive-using-java%2F&source=post_page---------------------------&ref=hackernoon.com)**Using Java** course.

You can also check out this list of[**30 linked list interview questions**](http://javarevisited.blogspot.sg/2017/07/top-10-linked-list-coding-questions-and.html?source=post_page---------------------------&ref=hackernoon.com) for more practice questions.

**3. String Coding Interview Questions**

Along with array and linked list data structures, a string is another popular  
topic on programming job interviews. I have never participated in a  
coding interview where no [string-based questions](http://www.java67.com/2018/04/21-string-programming-and-coding-interview-questions-answers.html?source=post_page---------------------------&ref=hackernoon.com) were asked.

A good thing about the string is that if you know the array, you can solve string-based questions easily because **strings are nothing but a character array**.

So all the techniques you learn by solving array-based coding questions  
can be used to solve string programming questions as well.

Here is my list of frequently asked string coding questions from programming job interviews:

1. **How do you print duplicate characters from a string? (**[**solution**](http://java67.blogspot.sg/2014/03/how-to-find-duplicate-characters-in-String-Java-program.html?source=post_page---------------------------&ref=hackernoon.com)**)**
2. **How do you check if two strings are anagrams of each other? (**[**solution**](http://javarevisited.blogspot.sg/2013/03/Anagram-how-to-check-if-two-string-are-anagrams-example-tutorial.html?source=post_page---------------------------&ref=hackernoon.com)**)**
3. **How do you print the first non-repeated character from a string? (**[**solution**](http://javarevisited.blogspot.sg/2014/03/3-ways-to-find-first-non-repeated-character-String-programming-problem.html?source=post_page---------------------------&ref=hackernoon.com)**)**
4. **How can a given string be reversed using recursion? (**[**solution**](http://javarevisited.blogspot.sg/2012/01/how-to-reverse-string-in-java-using.html?source=post_page---------------------------&ref=hackernoon.com)**)**
5. **How do you check if a string contains only digits? (**[**solution**](http://javarevisited.blogspot.sg/2012/10/regular-expression-example-in-java-to-check-String-number.html?source=post_page---------------------------&ref=hackernoon.com)**)**
6. **How are duplicate characters found in a string? (**[**solution**](http://java67.blogspot.sg/2014/03/how-to-find-duplicate-characters-in-String-Java-program.html?source=post_page---------------------------&ref=hackernoon.com)**)**
7. **How do you count a number of vowels and consonants in a given string? (**[**solution**](http://java67.blogspot.sg/2013/11/how-to-count-vowels-and-consonants-in-Java-String-word.html?source=post_page---------------------------&ref=hackernoon.com)**)**
8. **How do you count the occurrence of a given character in a string? (**[**solution**](http://javarevisited.blogspot.sg/2012/12/how-to-count-occurrence-of-character-in-String.html?source=post_page---------------------------&ref=hackernoon.com)**)**
9. **How do you find all permutations of a string? (**[**solution**](http://javarevisited.blogspot.com/2015/08/how-to-find-all-permutations-of-string-java-example.html?source=post_page---------------------------&ref=hackernoon.com)**)**
10. **How do you reverse words in a given sentence without using any library method? (**[**solution**](http://java67.blogspot.com/2015/06/how-to-reverse-words-in-string-java.html?source=post_page---------------------------&ref=hackernoon.com)**)**
11. **How do you check if two strings are a rotation of each other? (**[**solution**](http://www.java67.com/2017/07/string-rotation-in-java-write-program.html?source=post_page---------------------------&ref=hackernoon.com)**)**
12. **How do you check if a given string is a palindrome? (**[**solution**](http://java67.blogspot.com/2015/06/how-to-check-is-string-is-palindrome-in.html?source=post_page---------------------------&ref=hackernoon.com)**)**

These questions help improve your knowledge of string as a data structure. If you can solve all these String questions without any help then you are in good shape.

For more advanced questions, I suggest you solve problems given in the[**Algorithm Design Manual by Steven Skiena**](http://www.amazon.com/Algorithm-Design-Manual-Steven-Skiena/dp/1849967202?tag=javamysqlanta-20&source=post_page---------------------------&ref=hackernoon.com), a book with the toughest algorithm questions.

If you need more practice, here is another list of[**20 string coding questions**](http://javarevisited.blogspot.sg/2015/01/top-20-string-coding-interview-question-programming-interview.html?source=post_page---------------------------&ref=hackernoon.com).

**4. Binary Tree Coding Interview Questions**

So far, we have looked at only the linear data structure, but all  
information in the real world cannot be represented in linear fashion,  
and that’s where tree data structure helps.

Tree data structure is a data structure that allows you to store your data in a hierarchical fashion. Depending on how you store data, there are different types of trees, such as a [binary tree](http://javarevisited.blogspot.sg/2016/07/binary-tree-preorder-traversal-in-java-using-recursion-iteration-example.html?source=post_page---------------------------&ref=hackernoon.com), where each node has, at most, two child nodes.a

Along with its close cousin [binary search tree](http://javarevisited.blogspot.sg/2017/04/recursive-binary-search-algorithm-in-java-example.html?source=post_page---------------------------&ref=hackernoon.com), it’s also one of the most popular tree data structures. Therefore, you will find a lot of questions based on them, such as how to traverse them, count nodes, find depth, and check if they are balanced or not.

A key point to solving binary tree questions is a strong knowledge of theory, e.g. what is the size or depth of the binary tree, what is a leaf, and  
what is a node, as well as an understanding of the popular traversing  
algorithms, e.g. pre-, post-, and in-order traversal.

Here is a list of popular binary tree-based coding questions from software engineer or developer job interviews:

1. **How is a binary search tree implemented? (**[**solution**](http://javarevisited.blogspot.sg/2015/10/how-to-implement-binary-search-tree-in-java-example.html?source=post_page---------------------------&ref=hackernoon.com#axzz4wnEtnNB3)**)**
2. **How do you perform preorder traversal in a given binary tree? (**[**solution**](http://javarevisited.blogspot.sg/2016/07/binary-tree-preorder-traversal-in-java-using-recursion-iteration-example.html?source=post_page---------------------------&ref=hackernoon.com#axzz5ArdIFI7y)**)**
3. **How do you traverse a given binary tree in preorder without recursion? (**[**solution**](http://www.java67.com/2016/07/binary-tree-preorder-traversal-in-java-without-recursion.html?source=post_page---------------------------&ref=hackernoon.com)**)**
4. **How do you perform an inorder traversal in a given binary tree? (**[**solution**](http://www.java67.com/2016/08/binary-tree-inorder-traversal-in-java.html?source=post_page---------------------------&ref=hackernoon.com)**)**
5. **How do you print all nodes of a given binary tree using inorder traversal without recursion? (**[**solution**](http://www.java67.com/2016/08/binary-tree-inorder-traversal-in-java.html?source=post_page---------------------------&ref=hackernoon.com)**)**
6. **How do you implement a postorder traversal algorithm? (**[**solution**](http://www.java67.com/2016/10/binary-tree-post-order-traversal-in.html?source=post_page---------------------------&ref=hackernoon.com)**)**
7. **How do you traverse a binary tree in postorder traversal without recursion? (**[**solution**](http://www.java67.com/2017/05/binary-tree-post-order-traversal-in-java-without-recursion.html?source=post_page---------------------------&ref=hackernoon.com)**)**
8. **How are all leaves of a binary search tree printed? (**[**solution**](http://www.java67.com/2016/09/how-to-print-all-leaf-nodes-of-binary-tree-in-java.html?source=post_page---------------------------&ref=hackernoon.com)**)**
9. **How do you count a number of leaf nodes in a given binary tree? (**[**solution**](http://javarevisited.blogspot.sg/2016/12/how-to-count-number-of-leaf-nodes-in-java-recursive-iterative-algorithm.html?source=post_page---------------------------&ref=hackernoon.com)**)**
10. **How do you perform a binary search in a given array? (**[**solution**](http://javarevisited.blogspot.sg/2015/10/how-to-implement-binary-search-tree-in-java-example.html?source=post_page---------------------------&ref=hackernoon.com#axzz4wnEtnNB3)**)**

If you feel that your understanding of binary tree coding is inadequate  
and you can’t solve these questions on your own, I suggest you go back  
and pick a good data structure and algorithm course like [**From 0 to 1: Data Structures & Algorithms in Java**](https://click.linksynergy.com/fs-bin/click?id=JVFxdTr9V80&subid=0&offerid=323058.1&type=10&tmpid=14538&RD_PARM1=https%3A%2F%2Fwww.udemy.com%2Ffrom-0-to-1-data-structures%2F&source=post_page---------------------------&ref=hackernoon.com).

If you need some more recommendations, here is my list of useful [data structure algorithm books](http://javarevisited.blogspot.sg/2015/07/5-data-structure-and-algorithm-books-best-must-read.html?source=post_page---------------------------&ref=hackernoon.com) and [courses](http://javarevisited.blogspot.sg/2018/01/top-5-free-data-structure-and-algorithm-courses-java--c-programmers.html?source=post_page---------------------------&ref=hackernoon.com) to start with.

**5. Miscellaneous Coding Interview Questions**

Apart from data structure-based questions, most of the programming job  
interviews also ask algorithm, design, bit manipulation, and general  
logic-based questions, which I’ll describe in this section.

It’s important that you practice these concepts because sometimes they  
become tricky to solve in the actual interview. Having practiced them  
before not only makes you familiar with them but also gives you more  
confidence in explaining the solution to the interviewer.

1. **How is a bubble sort algorithm implemented? (**[**solution**](http://javarevisited.blogspot.sg/2014/08/bubble-sort-algorithm-in-java-with.html?source=post_page---------------------------&ref=hackernoon.com#axzz5ArdIFI7y)**)**
2. **How is an iterative quicksort algorithm implemented? (**[**solution**](http://javarevisited.blogspot.sg/2016/09/iterative-quicksort-example-in-java-without-recursion.html?source=post_page---------------------------&ref=hackernoon.com#axzz5ArdIFI7y)**)**
3. **How do you implement an insertion sort algorithm? (**[**solution**](http://www.java67.com/2014/09/insertion-sort-in-java-with-example.html?source=post_page---------------------------&ref=hackernoon.com)**)**
4. **How is a merge sort algorithm implemented? (**[**solution**](http://www.java67.com/2018/03/mergesort-in-java-algorithm-example-and.html?source=post_page---------------------------&ref=hackernoon.com)**)**
5. **How do you implement a bucket sort algorithm? (**[**solution**](http://javarevisited.blogspot.sg/2017/01/bucket-sort-in-java-with-example.html?source=post_page---------------------------&ref=hackernoon.com)**)**
6. **How do you implement a counting sort algorithm? (**[**solution**](http://www.java67.com/2017/06/counting-sort-in-java-example.html?source=post_page---------------------------&ref=hackernoon.com)**)**
7. **How is a radix sort algorithm implemented? (**[**solution**](http://www.java67.com/2018/03/how-to-implement-radix-sort-in-java.html?source=post_page---------------------------&ref=hackernoon.com)**)**
8. **How do you swap two numbers without using the third variable? (**[**solution**](http://www.java67.com/2015/08/how-to-swap-two-integers-without-using.html?source=post_page---------------------------&ref=hackernoon.com)**)**
9. **How do you check if two rectangles overlap with each other? (**[**solution**](http://javarevisited.blogspot.sg/2016/10/how-to-check-if-two-rectangle-overlap-in-java-algorithm.html?source=post_page---------------------------&ref=hackernoon.com)**)**
10. **How do you design a vending machine? (**[**solution**](http://javarevisited.blogspot.sg/2016/06/design-vending-machine-in-java.html?source=post_page---------------------------&ref=hackernoon.com)**)**