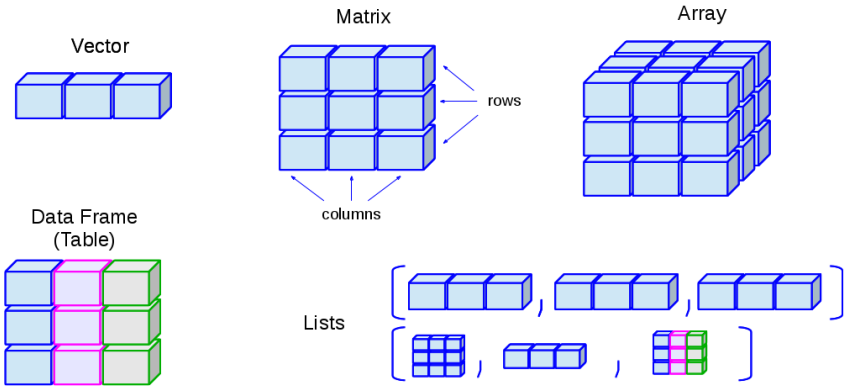
Interview Questions for Programmers

javinpaul Follow
Sep 23, 2018 · 11 min read



There are a lot of computer science graduates and programmers applying for programming, coding, and software development roles at startups like Uber and Netflix; big organizations like Amazon, Microsoft, and Google; and service-based companies like Infosys or Luxsoft, but many of them have no idea of **what kind of programming interview questions to expect** when you're applying for a job with these companies.

In this article, I'll share some *frequently asked programming interview questions* from different interviews for programmers at different levels of experience, **from people who have just graduated from college to programmers with one to two years of experience**.

Coding interviews are comprised mainly of data structure and algorithm-based questions as well as some of the logical questions such as, *How do you swap two integers without using a temporary variable?*

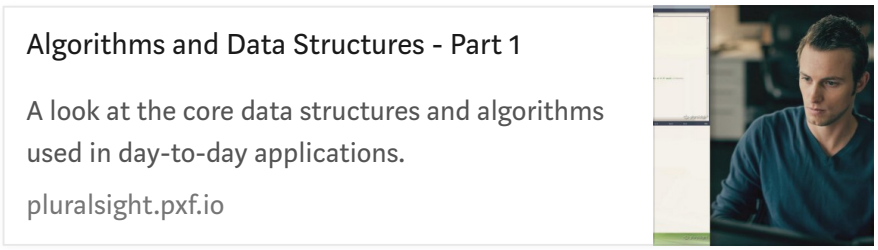
I think it's helpful to divide coding interview questions into different topic areas. The topic areas I've seen most often in interviews are array, linked list, string, binary tree, as well as questions from algorithms (e.g. string algorithm, sorting algorithms like quicksort or radix sort, and other miscellaneous ones), and that's what you will find in this article.

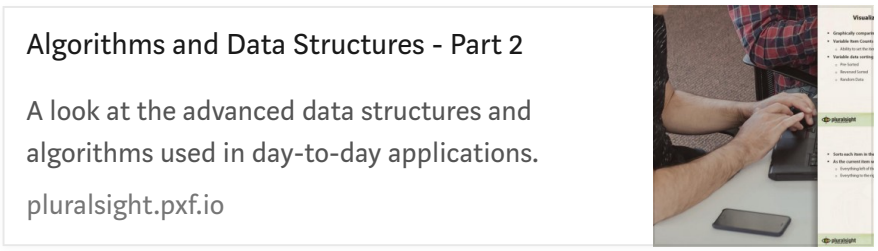
It's **not guaranteed** that you will be asked these coding or data structure and algorithmic questions, but they will give you enough of an idea of the kinds of questions you can expect in a real programming job interview.

Once you have gone through these questions, you should feel confident enough to attend any telephonic or face-to-face interviews.

Btw, there is no point in attempting these questions if you don't have sufficient knowledge of **essential Data Structure and Algorithms** or you have not touched them from ages.

In that case, you should take a good course like **Algorithms and Data Structures Part 1 and 2 By Robert Horvick** to refresh your DS and algorithms skills.





Top 50 Algorithms and Coding Interview Questions

Without any further ado, here is my list of some of the **most frequently asked coding interview questions from programming job interviews**:

1. Array Coding Interview Questions

An array is the most fundamental data structure, which stores elements at a contiguous memory location. It is also one of the darling topics of interviewers and you will hear a lot of questions about an array in any

coding interview, e.g. reversing an array, sorting the array, or searching elements on the array.

The key benefit of an array data structure is that it offers fast O(1) search if you know the index, but **adding and removing an element from an array is slow** because you cannot change the size of the array once it's created.

In order to create a shorter or longer array, you need to create a new array and copy all elements from old to new.

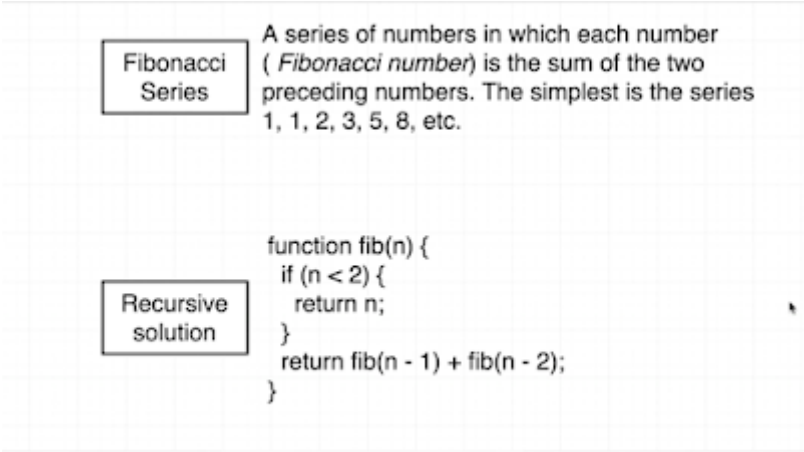
The key to solving array-based questions is having a good knowledge of array data structure as well as basic programming constructors such as loop, recursion, and fundamental operators.

Here are some of the popular array-based coding interview questions for your practice:

1. **How do you find the missing number in a given integer array of 1 to 100? (solution)**
2. **How do you find the duplicate number on a given integer array? (solution)**
3. **How do you find the largest and smallest number in an unsorted integer array? (solution)**
4. **How do you find all pairs of an integer array whose sum is equal to a given number? (solution)**
5. **How do you find duplicate numbers in an array if it contains multiple duplicates? (solution)**
6. **How are duplicates removed from a given array in Java? (solution)**
7. **How is an integer array sorted in place using the quicksort algorithm? (solution)**
8. **How do you remove duplicates from an array in place? (solution)**
9. **How do you reverse an array in place in Java? (solution)**
10. **How are duplicates removed from an array without using any library? (solution)**

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**, 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**.

2. Linked List Programming Interview Questions

A linked list 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 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 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)**
2. **How do you check if a given linked list contains a cycle? How do you find the starting node of the cycle? (solution)**
3. **How do you reverse a linked list? (solution)**
4. **How do you reverse a singly linked list without recursion? (solution)**
5. **How are duplicate nodes removed in an unsorted linked list? (solution)**
6. **How do you find the length of a singly linked list? (solution)**
7. **How do you find the third node from the end in a singly linked list? (solution)**
8. **How do you find the sum of two linked lists using Stack? (solution)**

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 Using Java** course.

You can also check out this list of **30 linked list interview questions** 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 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.

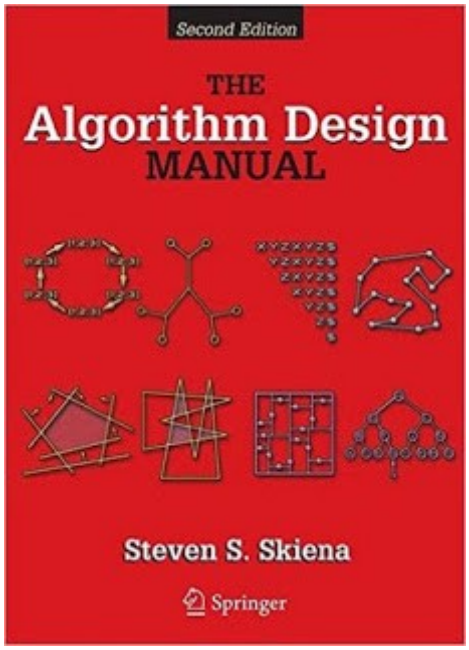
Here is my list of frequently asked string coding questions from programming job interviews:

1. **How do you print duplicate characters from a string? (solution)**
2. **How do you check if two strings are anagrams of each other? (solution)**
3. **How do you print the first non-repeated character from a string? (solution)**
4. **How can a given string be reversed using recursion? (solution)**
5. **How do you check if a string contains only digits? (solution)**
6. **How are duplicate characters found in a string? (solution)**

7. **How do you count a number of vowels and consonants in a given string? (solution)**
8. **How do you count the occurrence of a given character in a string? (solution)**
9. **How do you find all permutations of a string? (solution)**
10. **How do you reverse words in a given sentence without using any library method? (solution)**
11. **How do you check if two strings are a rotation of each other? (solution)**
12. **How do you check if a given string is a palindrome? (solution)**

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**, a book with the toughest algorithm questions.



If you need more practice, here is another list of **20 string coding questions**.

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, where each node has, at most, two child nodes.a

Along with its close cousin binary search tree, 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.

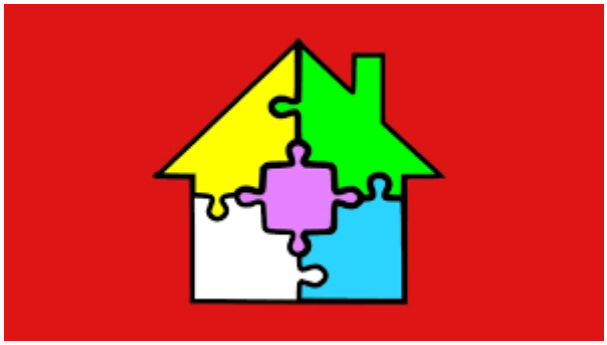
A key point to solving binary tree questions is a strong knowledge of theory, e.g. what is the size or depth of the binary tree, what is a leaf, and what is a node, as well as an understanding of the popular traversing algorithms, e.g. pre-, post-, and in-order traversal.

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)**
2. **How do you perform preorder traversal in a given binary tree? (solution)**
3. **How do you traverse a given binary tree in preorder without recursion? (solution)**
4. **How do you perform an inorder traversal in a given binary tree? (solution)**
5. **How do you print all nodes of a given binary tree using inorder traversal without recursion? (solution)**
6. **How do you implement a postorder traversal algorithm? (solution)**
7. **How do you traverse a binary tree in postorder traversal without recursion? (solution)**
8. **How are all leaves of a binary search tree printed? (solution)**
9. **How do you count a number of leaf nodes in a given binary tree? (solution)**

10. **How do you perform a binary search in a given array? (solution)**

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**.



If you need some more recommendations, here is my list of useful data structure algorithm books and courses to start with.

5. Miscellaneous Coding Interview Questions

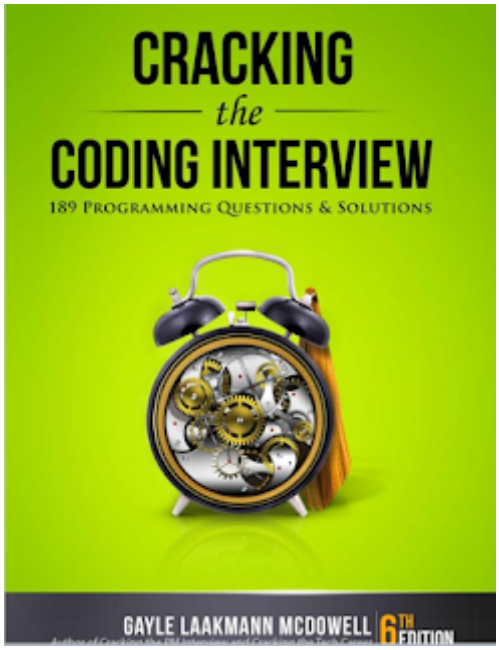
Apart from data structure-based questions, most of the programming job interviews also ask algorithm, design, bit manipulation, and general logic-based questions, which I'll describe in this section.

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)**
2. **How is an iterative quicksort algorithm implemented? (solution)**
3. **How do you implement an insertion sort algorithm? (solution)**
4. **How is a merge sort algorithm implemented? (solution)**
5. **How do you implement a bucket sort algorithm? (solution)**
6. **How do you implement a counting sort algorithm? (solution)**

7. **How is a radix sort algorithm implemented? (solution)**
8. **How do you swap two numbers without using the third variable? (solution)**
9. **How do you check if two rectangles overlap with each other? (solution)**
10. **How do you design a vending machine? (solution)**

If you need more such coding questions you can take help from books like **Cracking The Code Interview, by Gayle Laakmann McDowell** which presents **189+ Programming questions and solution.** A good book to prepare for programming job interviews in a short time.



By the way, the more questions you solve in practice, the better your preparation will be. So, if you think 50 is not enough and you need more, then check out these additional **50 programming questions** for telephone interviews and these books and courses for a more thorough preparation.

Now You're Ready for the Coding Interview

These are some of the most common questions outside of data structure and algorithms that help you to do really well in your interview.

I have also shared a lot of these questions on my blog, so if you are really interested, you can always go there and search for them.

These **common coding, data structure, and algorithm questions** are the ones you need to know to successfully interview with any company, big or small, for any level of programming job.

If you are looking for a programming or software development job in 2018, you can start your preparation with this list of coding questions.

This list provides good topics to prepare and also helps assess your preparation to find out your areas of strength and weakness.

A good knowledge of data structure and algorithms is important for success in coding interviews and that's where you should focus most of your attention.

Further Learning
Data Structures and Algorithms: Deep Dive Using Java
10 Books to Prepare Technical Programming/Coding Job Interviews
10 Algorithm Books Every Programmer Should Read
Top 5 Data Structure and Algorithm Books for Java Developers
From 0 to 1: Data Structures & Algorithms in Java
Data Structure and Algorithms Analysis—Job Interview

Closing Notes

Thanks, You made it to the end of the article … Good luck with your programming interview! It's certainly not going to be easy, but by following this roadmap and guide, you are one step closer to becoming a DevOps engineer.

If you like this article, then please share with your friends and colleagues, and don't forget to follow javinpaul on Twitter!

P.S.—If you need some FREE resources, you can check out this list of free data structure and algorithm courses to start your preparation.