64 Coding Interview Questions + Answers [2024 Prep Guide]

Technical Coding Interview Questions [Programming-Based]

We’ll start with the common questions that are used to evaluate your foundational knowledge of coding. Depending on the interview, you may be expected to present your answers vocally, write them on a board, or use a provided laptop.

String

The following questions assess your understanding of the “string” data type.

How Do You Reverse a String?

There are many methods available to reverse a string in Python, for example:

* Slicing
* Use join
* Loop

The below [Python](https://www.springboard.com/blog/data-science/python-guide/) snippet shows an example of the loop method being used to iterate characters in reverse order and then append them.

def reverse(myString):

reversedString = ''

index = len(myString)

while index:

index -= 1

reversedString += myString[index]

return reversedString

How Do You Calculate the Number of Vowels and Consonants in a String?

1. Create a lower-case copy of the string to simplify the checks.
2. Loop through each character in the string.
3. Check against each vowel, as the set is small.
4. If it’s not a vowel, check if it is within the [ASCII table values of all known letters](https://www.asciitable.com/mobile/) (as a character is ultimately a numerical value).

void CountVowelsAndConsonants(string input, out int numVowels, out int numConsonants)

{

numVowels = 0;

numConsonants = 0;

var loweredInput = input.ToLower();

foreach (var c in loweredInput)

{

if (c == 'a' || c == 'e' || c == 'i' || c == 'u')

numVowels++;

else if (c >= 'a' && c <= 'z')

numConsonants++;

}

How Do You Print the First Non-Repeated Character in a String?

1. Create an array of 256, the amount of valid ASCII characters, and initialize each value to -1.
2. Loop through each character in the input string.
3. Using the character as an index in the indices array, if the value is -1, you are seeing it for the first time. In that case, set it to the current loop index.
4. If the value isn’t -1, you have processed this character before, so mark this with -2.
5. Loop through the indices array and find the smallest value (this is the index in the string), this is your first non-repeating character.

int[] indices = new int[256];

for (int i = 0; i < 256; i++)

indices[i] = -1;

for(int i = 0; i < input.Length; i++)

{

var c = input[i];

if (indices[c] == -1)

indices[c] = i;

else

indices[c] = -2;

}

int firstNonRepeatingIndex = Int32.MaxValue;

foreach (var index in indices)

if (index >= 0)

firstNonRepeatingIndex = Math.Min(firstNonRepeatingIndex, index);

How Do You Figure Out if the Provided String Is a Palindrome?

A palindrome is a string of characters that retains the same order whether it is written forward or backward. You can find out if a string is a palindrome by reversing it and then comparing the two versions as shown below:

bool IsPalindrome(string input)

{

var inputCharArray = input.ToCharArray();

Array.Reverse(inputCharArray);

var reversedInput = new string(inputCharArray);

var isPalindrome = string.Equals(reversedInput, input, StringComparison.CurrentCultureIgnoreCase);

return isPalindrome;

}

How Do You Determine Whether the Following Two Strings Are Anagrams of Each Other?

Re-order the strings alphabetically and then compare the two strings.

string a = "abcd";

string b = "bcda";

bool AreAnagrams(string inputA, string inputB)

{

// We reorder the strings alphabetically

var orderedA = String.Concat(inputA.ToLower().OrderBy(c => c));

var orderedB = String.Concat(inputB.ToLower().OrderBy(c => c));

// Now if they contain the same letters, they should be the exact same string

return orderedA == orderedB;

}

Console.WriteLine(AreAnagrams(a, b));

Array

The next section covers common questions related to arrays.

How Do You Find Duplicate Numbers in an Array With Multiple Duplicates?

1. Create a dictionary where the number will be the key, and then set the number of times it appears as the value.
2. Loop through the array and increment the count for the key entry in the dictionary.
3. Finally, loop through all pairs in the dictionary; if the value is more than 1, it is duplicated in the array.

int[] array = { 5, 2, 1, 5, 8, 5, 1, 7, 7, 0 };

var duplicateDict = new Dictionary<int, int>();

foreach(var val in array)

{

duplicateDict.TryGetValue(val, out var count);

duplicateDict[val] = count + 1;

}

foreach (var pair in duplicateDict)

{

if (pair.Value > 1)

{

Console.WriteLine("Value {0} was duplicated and occurred {1} times.", pair.Key, pair.Value);

}

}

How Do You Find the Largest and Smallest Number in an Array of 1–100?

1. Generate an array of all integers from 1 to 100.
2. Have a variable for the largest and smallest numbers you have found so far, initialized to the smallest and largest integer, respectively.
3. Loop through each value in the array and use Max and Min to assign if it is larger or smaller than the numbers you already have stored.
4. These values contain the largest and smallest numbers in the array.

// Generate an array of ints containing the numbers from 1 to 100

var array = new int[100];

for (int i = 0; i < 100; i++)

array[i] = i + 1;

int largestNumber = Int32.MinValue;

int smallestNumber = Int32.MaxValue;;

foreach (var number in array)

{

largestNumber = Math.Max(largestNumber, number);

smallestNumber = Math.Min(smallestNumber, number);

}

Console.WriteLine($"Largest Number: {largestNumber}, Smallest Number: {smallestNumber}");

How Do You Remove Duplicates From an Array in Place?

“In Place” normally means without allocating any further memory. In this case, it is literally impossible to remove something “In Place,” as removing this would reduce the array size. Here’s how you can do this without a secondary array:

1. Sort the array by ascending value order. This will place duplicates next to each other.
2. Loop through the array from back to front. This is necessary to allow you to reduce it in size while iterating.
3. Track the current value until you find a different one while keeping track of the count of the same value up to this point.
4. If you find a new value or reach the start of the array, use the count to remove the duplicates using swap-back and resize. To do this, swap the unwanted value with the last value in the array and then resize the array down by 1.
5. All duplicates have been removed, but it is no longer a sorted array.

int[] array = { 5, 2, 1, 5, 8, 5, 1, 7, 7, 0 };

// First sort the array

Array.Sort(array);

int current = array[^1];

int duplicateCount = -1;

for(var i = array.Length-1; i >= 0; i--)

{

if (array[i] != current || i == 0)

{

// We found a new value or are at the end, so remove duplicates up to this point

int startIndex = i + 1;

for (int j = startIndex; j < startIndex+duplicateCount; j++)

{

array[j] ^= array[^1];

array[^1] ^= array[j];

array[j] ^= array[^1];

Array.Resize(ref array, array.Length - 1);

}

// Reset count and current

current = array[i];

duplicateCount = 0;

continue;

}

duplicateCount++;

}

foreach(var val in array)

Console.Write($"{val} ");

How Do You Find All Pairs in an Array of Integers That Add Up to a Specific Sum?

1. Loop through an array of values.
2. For each value, loop again through all other values.
3. Check if these two values add up to the provided sum and print them out if so.

int[] array = { 5, 2, 1, 5, 8, 5, 1, 7, 7, 0 };

void PrintPairs(int[] input, int sum)

{

for (int i = 0; i < input.Length; i++)

for (int j = i + 1; j < input.Length; j++)

if ((input[i] + input[j]) == sum)

Console.WriteLine($"Pair {input[i]}+{input[j]}=={sum}");

}

PrintPairs(array, 10);

Binary Tree

This section will cover [data structure interview questions](https://www.springboard.com/blog/software-engineering/data-structures-and-algorithms-interview-questions/) on binary trees.

How Do You Implement a Binary Search Tree?

A binary search tree is a tree data structure of nodes where each node contains a maximum of two linking nodes, usually known as Left and Right. In its simplest form, a binary search tree can therefore be implemented entirely as a single Node class containing the data and references to the Left and Right nodes.

public class Node

{

public int data;

public Node left, right;

public Node(int item)

{

data = item;

left = right = null;

}

}

var root = new Node(10);

root.left = new Node(8);

root.right = new Node(2);

How Do You Traverse a Given Binary Tree in Preorder Without Recursion?

Assuming you already have a binary tree implementation, as described above, follow these steps:

1. Use a stack to process depth-first.
2. Push the first node (usually the root) to the stack.
3. Use a while loop to process so long as the stack is not empty.
   1. Read the top node of the stack.
   2. Remove this from the stack.
   3. If this node has a Left or Right, add them to the stack.
4. The loop will keep adding nodes to the stack and processing them until the whole tree is processed, then the stack will be empty, and the loop will exit.

void IterativePreorder(Node node)

{

Stack<Node> nodeStack = new Stack<Node>();

nodeStack.Push(node);

while (nodeStack.Count > 0)

{

var topNode = nodeStack.Peek();

Console.Write(topNode.data + " ");

nodeStack.Pop();

if (topNode.right != null)

nodeStack.Push(topNode.right);

if (topNode.left != null)

nodeStack.Push(topNode.left);

}

}

Can You Count How Many Leaf Nodes Are in a Given Binary Tree?

A leaf node is simply a node that has no child nodes. You can use recursion to process the whole tree, and where a node has no Left or Right child nodes, you will add 1 to the leaf count. This will give you the total leaf count in the tree.

int CalcLeafCount(Node node)

{

if (node == null)

return 0;

if (node.left == null && node.right == null)

return 1;

return CalcLeafCount(node.left) + CalcLeafCount(node.right);

}

Write a Code To Find the Maximum Depth of a Binary Tree.

You can solve this with a code using a recursive function to process the whole tree, returning the calculated depth up the recursive call stack.

int CalcMaxDepth(Node node)

{

if (node == null)

return -1;

int leftDepth = CalcMaxDepth(node.left);

int rightDepth = CalcMaxDepth(node.right);

if (leftDepth > rightDepth)

return leftDepth + 1;

return rightDepth + 1;

}

Linked List

These next questions cover the linked list data structure.

How Do You Reverse a Linked List?

A linked list is comprised of nodes pointing to the previous and next nodes, so you can reverse a linked list by iterating the nodes in the list, and then swap the previous and next node entries.

void ReverseListRecursive(LinkedList linkedList)

{

linkedList.head = ReverseListRecursiveImpl(linkedList.head);

}

LinkedList.Node ReverseListRecursiveImpl(LinkedList.Node head)

{

if(head == null)

return head;

if(head.next == null)

return head;

LinkedList.Node newHead = ReverseListRecursiveImpl(head.next);

head.next.next = head;

head.next = null;

return newHead;

}

How Do You Reverse a Singly Linked List Without Recursion?

The principle of reversing a linked list without recursion is the same, and is in fact much simpler than the recursive solution. There is almost never a scenario where you should use recursion to iterate a linked list.

void ReverseList(LinkedList linkedList)

{

LinkedList.Node prev = null;

var current = linkedList.head;

while (current != null)

{

var next = current.next;

current.next = prev;

prev = current;

current = next;

}

linkedList.head = prev;

}

How Do You Determine Whether There Is a Loop in a Singly Linked List?

A loop (or cycle) in a linked list is when a node’s next value points back to an earlier node in the list.

1. Create a hash set of nodes. You will use this to track whether or not you have already processed a node.
2. Iterate the linked list, adding each node to the hash set.
3. If the node is already present in the hash set, then the linked list contains a loop (or cycle).

bool ContainsLoop(LinkedList linkedList)

{

HashSet<LinkedList.Node> nodeVisitSet = new HashSet<LinkedList.Node>();

var node = linkedList.head;

while (node != null)

{

// If it's there, we already visited so this is a loop

if (nodeVisitSet.Contains(node))

return true;

nodeVisitSet.Add(node);

node = node.next;

}

return false;

}

How Do You Add an Element to the Middle of a Linked List?

You can insert an element in the middle of a linked list by modifying the middle element to point to the new node. Next, modify the new node to point to the next node (which the middle node had previously pointed to).

void Insert(LinkedList.Node insertPos, LinkedList.Node newNode)

{

var oldNext = insertPos.next;

insertPos.next = newNode;

newNode.next = oldNext;

}

Other

The following section includes algorithm questions, Armstrong numbers, and Fibonacci numbers.

How Is a Bubble Sort Algorithm Implemented?

A bubble sort algorithm is implemented by iterating an array and repeatedly swapping pairs that are in the wrong order.

void BubbleSortArray(int[] array)

{

int n = array.Length;

for (int i = 0; i < n - 1; i++)

for (int j = 0; j < n - i - 1; j++)

{

if (array[j] > array[j + 1])

{

(array[j], array[j + 1]) = (array[j + 1], array[j]);

}

}

}

Write a Program To Find the Prime Factors of an Integer.

1. First, as long as the number is divisible by 2, print out 2 and divide it by 2.
2. Now that you have an odd number, loop from 3 to the square root of the number, incrementing by 2 each time.
3. While the number is divisible by i, print it out and divide by i.
4. Lastly, if the remaining number is larger than 2, print it out.

void PrintPrimeFactors(int number)

{

while (number % 2 == 0)

{

Console.Write(2 + " ");

number /= 2;

}

for (var i = 3; i <= Math.Sqrt(number); i+= 2)

{

while (number % i == 0)

{

Console.Write(i + " ");

number /= i;

}

}

if (number > 2)

Console.Write(number);

}

Check if the Following Number Is an Armstrong Number.

An Armstrong number is when the sum of the digits raised to the power of the number of digits equals the number itself. For example, 371 == 3^3 + 7^3 + 1^3.

bool IsArmstrongNumber(int number)

{

// Count the number of digits

int count = 0;

int numberCursor = number;

while (numberCursor > 0)

{

numberCursor /= 10;

count++;

}

// For each digit, raise it to the power of the digit count

// and sum them together

int sum = 0;

numberCursor = number;

while (numberCursor > 0)

{

sum += (int)Math.Pow(numberCursor % 10, count);

numberCursor /= 10;

}

// Check if the sum is the same as the number

return sum == number;

}

Write a Program To Print the First N Fibonacci Numbers.

Fibonacci numbers are numbers that equal the sum of the preceding two numbers. This is using an iterative method starting from 0 to N. For each Fibonacci number, add the previous two numbers to get the next number in the set.

void PrintFibonacci(int n)

{

var a = 0;

var b = 1;

for (var i = 0; i < n; i++)

{

var temp = a;

a = b;

b = temp + b;

Console.WriteLine(a);

}

}

PrintFibonacci(7);

How Do You Implement a Bucket Sort Algorithm?

1. Create a number of “buckets” for the length of the input array. Each bucket is a list of entries.
2. Iterate the input array, storing each value in a computed bucket index. The computed bucket index is the value of the current input array entry multiplied by the number of buckets.
3. Sort each bucket’s list.
4. Iterate all values in all buckets and write them back into the input array in this order.

void BucketSort(float[] array)

{

var numBuckets = array.Length;

var buckets = new List<float>[numBuckets];

for (var i = 0; i < numBuckets; i++)

buckets[i] = new List<float>();

for (var i = 0; i < numBuckets; i++)

{

var bucketIndex = array[i] \* numBuckets;

buckets[(int)bucketIndex].Add(array[i]);

}

for (var i = 0; i < numBuckets; i++)

buckets[i].Sort();

var arrayIndex = 0;

for (var i = 0; i < numBuckets; i++)

{

for (var j = 0; j < buckets[i].Count; j++)

{

array[arrayIndex++] = buckets[i][j];

}

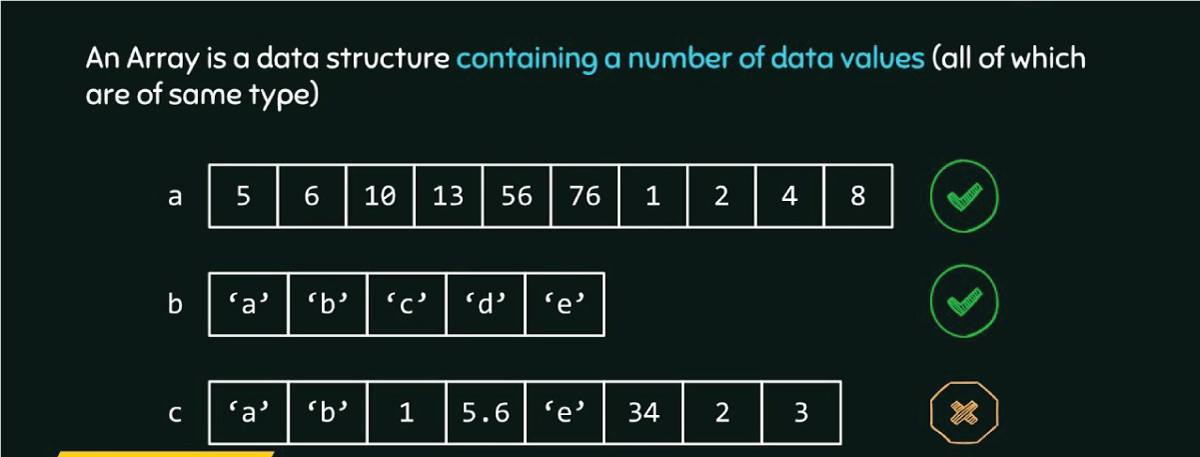
}

}

Coding Interview Questions [Concept-Based]

Concept-based questions evaluate your understanding of code-related concepts, as well as your ability to articulate this understanding.

What Is an Array?

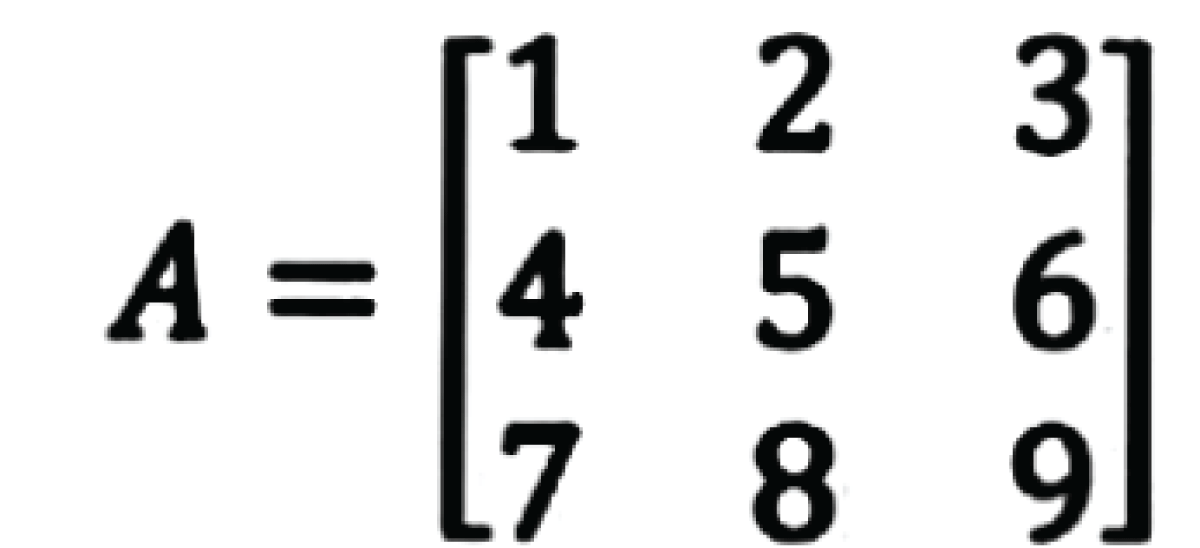
Source: [The Saas Way](http://thesassway.com/)

An array data structure is a group of similar data elements stored at adjacent memory locations, such as an integer array.

What Is a Stack?

A stack is an abstract data element and a collection of data elements. Elements can be added to the stack with the operation “push,” and removed with “pop.”

What Is a Matrix?

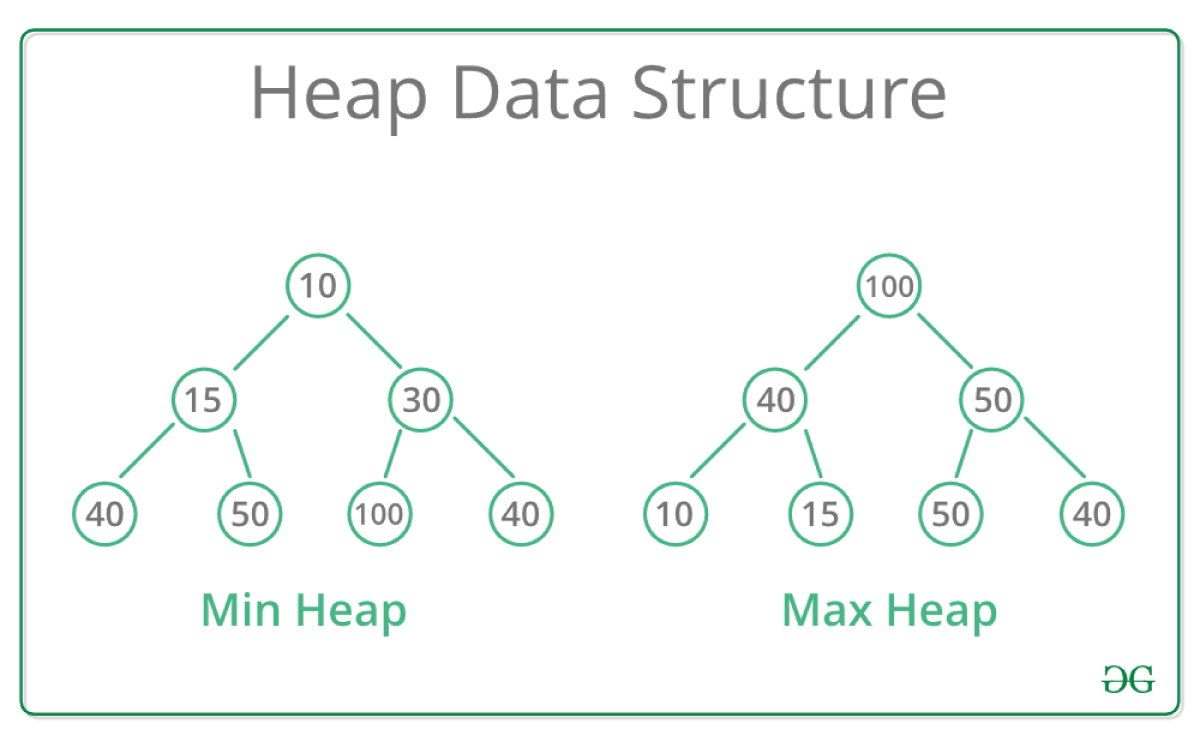
Source: [Study Pug](http://studypug.com/)

“Matrix” is a term used for a table of numbers made up of rows and columns.

What Is a Tree?

A tree is a non-linear data structure that allows for multiple relations between nodes.

What Is a Heap?

Source: [Greeks For Greeks](https://www.geeksforgeeks/)

A tree-based structure that works as a complete binary tree. There are two types of heaps: a min-heap and a max-heap. The root node of a min-heap must be the lowest value of all the nodes, and the root node of a max-heap must be the highest.

What Is a Queue?

A queue is a collection of entities that are kept in a specific sequence. Entities can be added to one end and removed from the other.

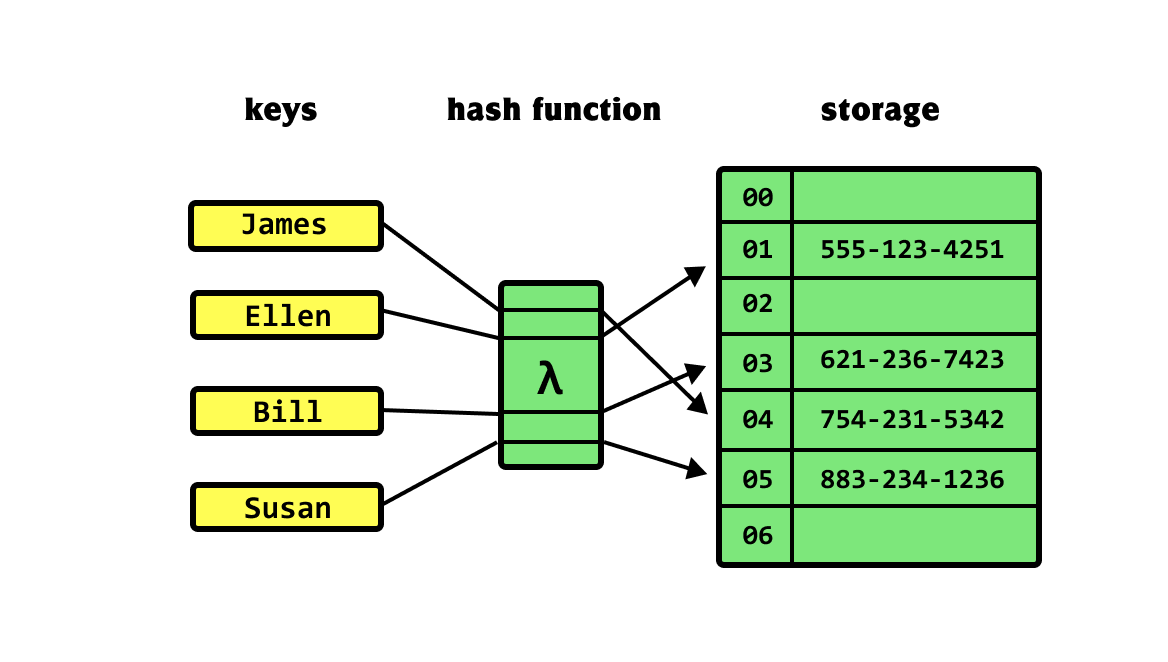
What Is a Graph?

A graph is a non-linear data structure made up of nodes and edges. The edges can connect any two nodes in the graph.

What Is a Linked List?

A linked list is a linear data structure where the entities are not stored next to each other on the memory. Instead, each node includes a reference to the following node in the list.

What Is a Hash Table?

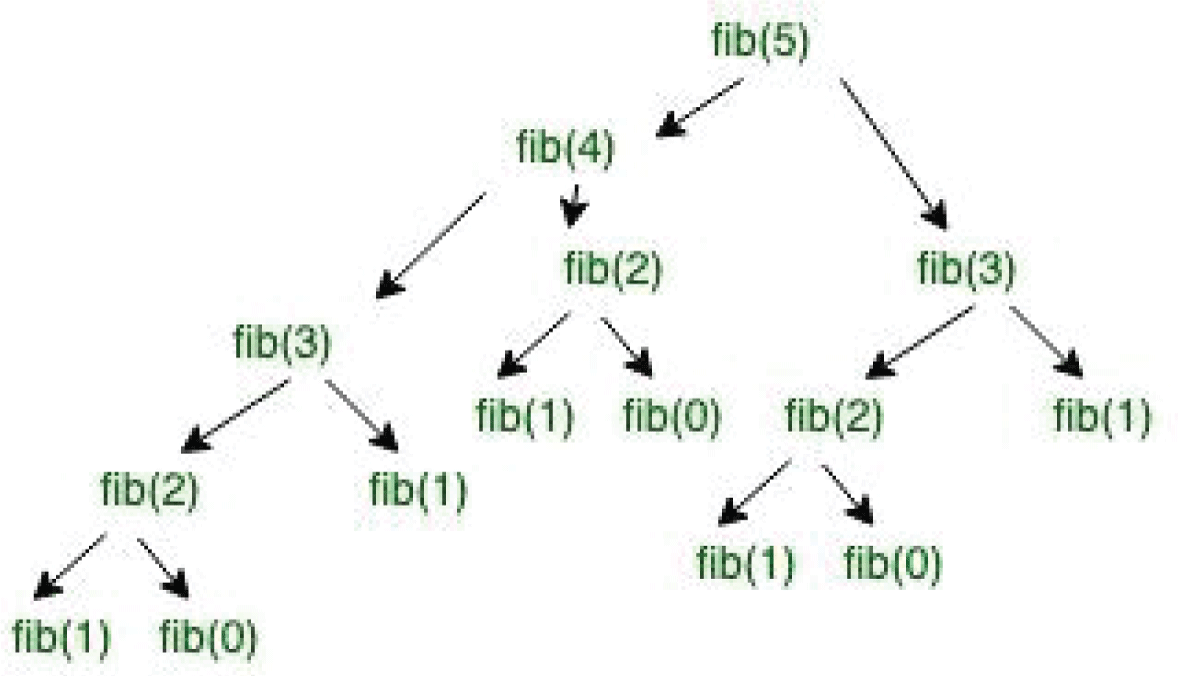
Source: [Khalil Stemmler](http://khalilstemmler.com/)

A hash table is a [data structure](https://www.springboard.com/blog/software-engineering/data-structures-and-algorithms/) that stores information through association. Each data element has a unique index value that identifies it and allows it to be accessed quickly.

What Is a Binary Search Tree?

A binary search tree uses the tree structure mentioned above. Every value in the left subtree is greater than the value of the root node, and every value in the right subtree is lower.

What Is Recursion?

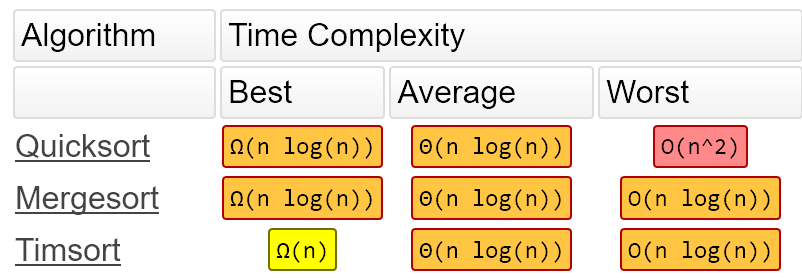
Source: [Greeks For Greeks](https://www.geeksforgeeks/)

Recursion is a way to solve problems that rely on looping code. Here’s how you do that:

1. Tell the computer to stop when the goal is reached.
2. Tell the computer to take a very simple step towards the goal.
3. Order the algorithm to be repeated.

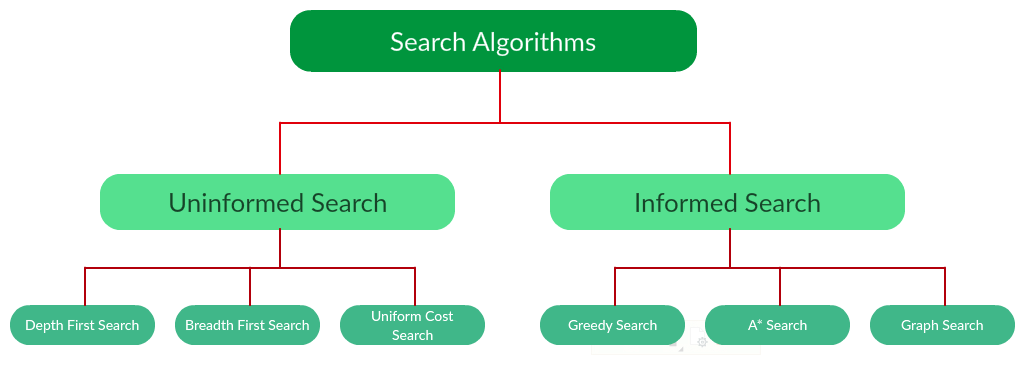
This way, the code can check whether the criteria defined have been met and if they haven’t, proceed to take a small step forward. The algorithm will then repeat until the criteria are all met and the second step is not required.

What Is a Sorting Algorithm?

Source: [Free Code Camp](http://freecodecamp.org/)

A sorting algorithm uses a set collection of instructions to sort the items of an array or list into a certain order. This is most commonly a numerical or alphabetical order.

What Is a Searching Algorithm?

Source: [Section](http://section.io/)

A set of instructions that confirms the existence of or retrieves certain data within a larger group of data.

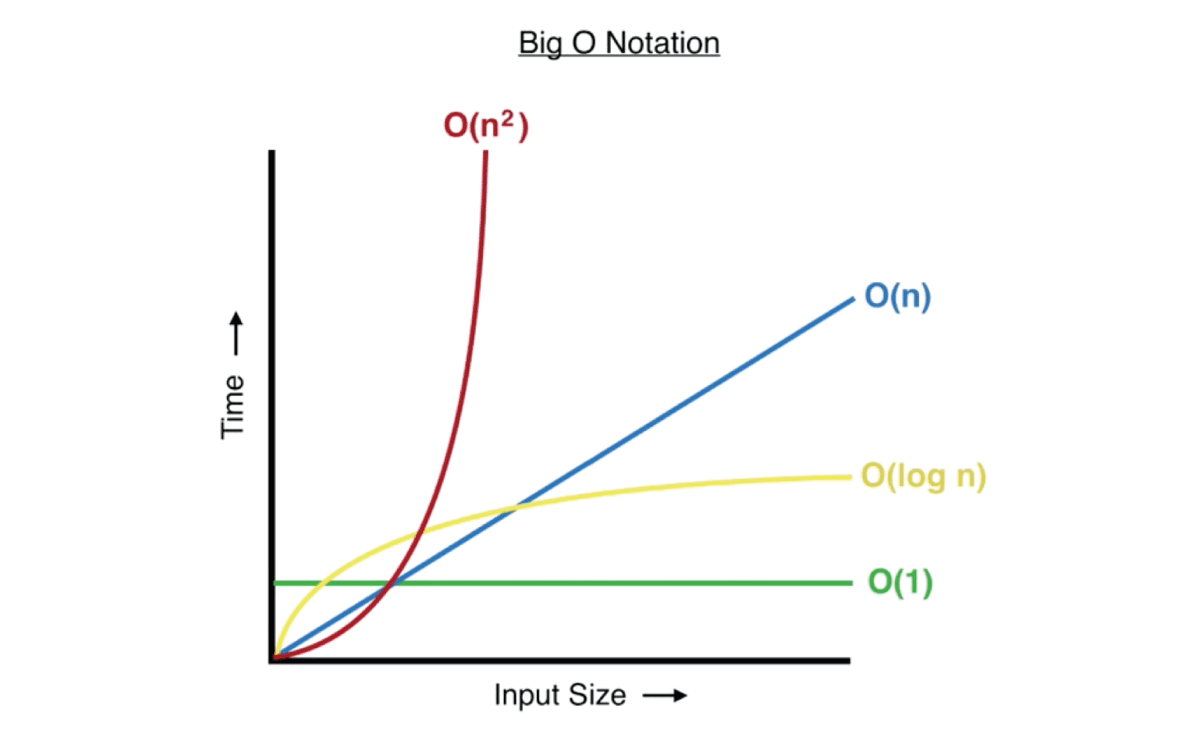
What Is a Hashing Algorithm?

A hashing algorithm assigns a fixed value to a data element, allowing it to be used in a hash table.

What Is a Greedy Algorithm?

A greedy algorithm approaches each stage of a problem in isolation, selecting the most optimal solution for that particular stage without considering any other factors.

What Is the “Big O” Notation?

Source: [Medium Dataseries](https://medium.com/dataseries)

The “Big O” is a mathematical notation that measures the efficiency of an algorithm by conserving the amount of time it takes and the amount of space it uses. It always measures based on the worst-case scenario.

Do You Think OOP Is a Good Coding Method?

Object-oriented programming (OOP) has been the [dominant paradigm for decades](https://prateekvjoshi.com/2013/11/30/programming-paradigms-object-oriented-vs-data-oriented/), though it is debated whether this was due to merit or chance. In more recent years, data-oriented programming (DOP) has also gained popularity and can be used to great effect in many areas. However, rather than saying one paradigm is better than the other, it is more accurate to say that there are pros and cons to both.

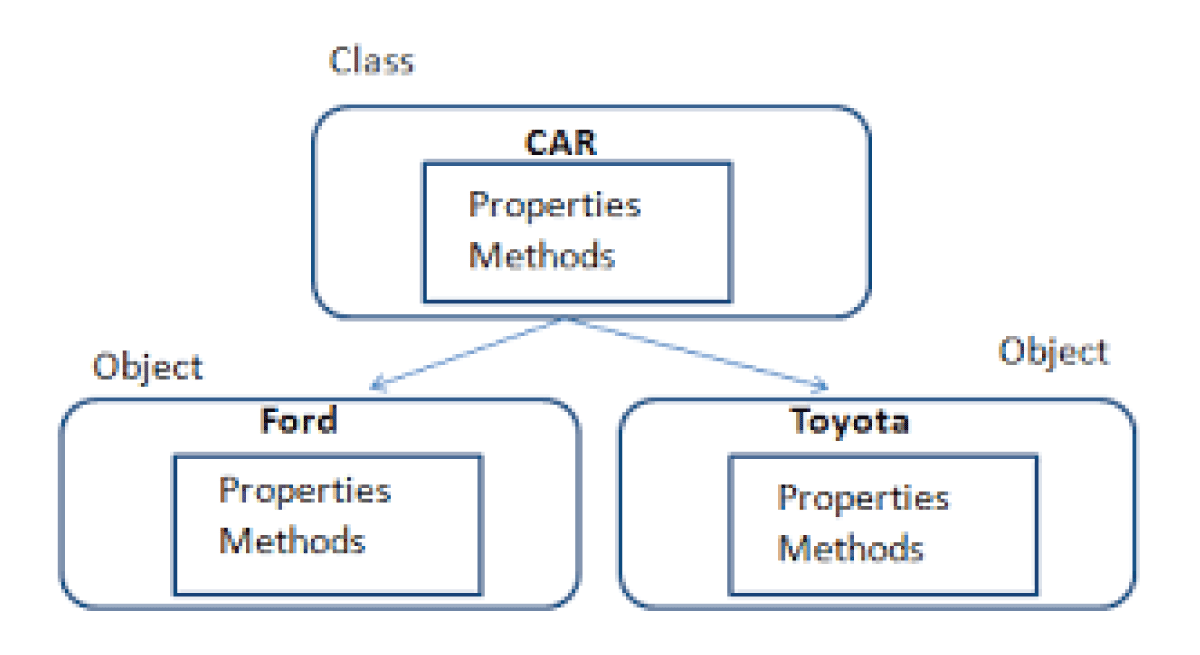
What Are Leaf Nodes and Why Are They Important in a Binary Tree?

Nodes in a binary tree that have children are referred to as interior nodes. When a node has no children (and therefore is a dead end), it’s called a leaf node. A leaf node signifies the end of a branch and lets the computer know that it doesn’t need to be processed any further.

What Is the Difference Between a Programming Language and a Script?

The main difference between the two is that a programming language is compiled before running, meaning it is converted to machine code that the computer can directly read. Scripting languages, however, are instead interpreted by the computer in real-time.

What Is the Difference Between a Class and an Object?

Source: [Net Informations](http://net-informations.com/)

A class defines the properties of an object, and every object fits within a class. In other words, classes are categories, and objects are sorted into categories.

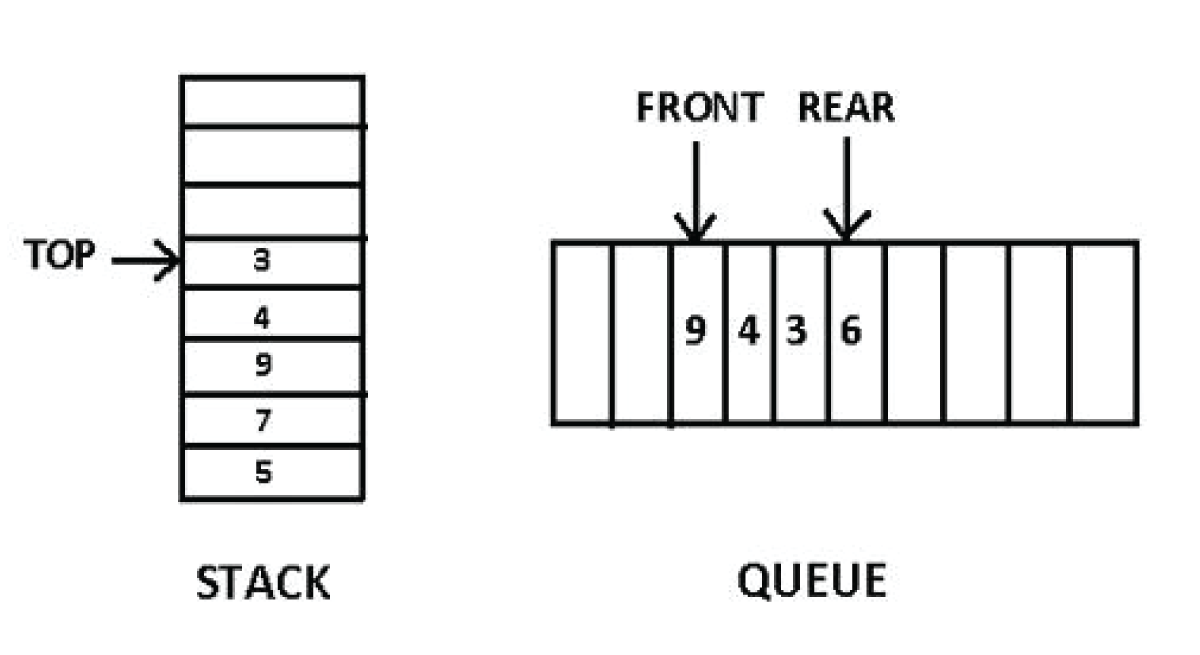
What Is the Difference Between an Interpreter and a Compiler?

A compiler converts an entire program into machine code before running it, whereas an interpreter changes each line one by one as it runs the program.

What Is the Difference Between an Array and a Linked List?

Both are collections of similar data types, but they differ in the way the data is stored. An array stores data in adjacent locations in the memory, whereas each element in a linked list includes a reference to the following element.

What Is the Difference Between a Stack and a Queue?

Source: [QS Study](http://qsstudy.com/)

Both a stack and a queue are ordered lists of data, but while queues can have elements added or removed from both ends, a stack only allows it to happen from one end.

What Is the Difference Between a String and a StringBuilder in Java?

A string in Java is known as “immutable.” This means its contents cannot be altered after it has been created. A StringBuilder, however, can be changed.

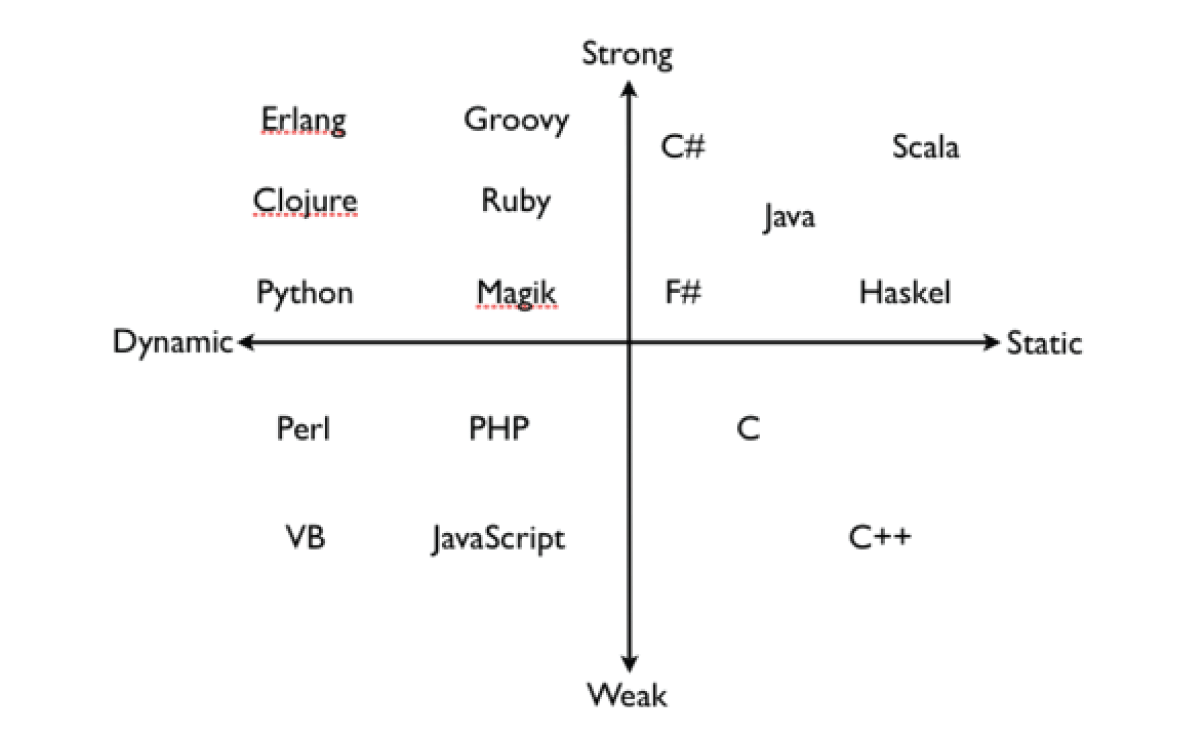
What Is the Difference Between a Breadth-First and a Depth-First Search?

When searching a binary tree, the depth-first approach will follow the nodes downward and process them in that order. In comparison, a breadth-first search will process all adjacent nodes before moving downward.

What’s the Difference Between Searching and Sorting?

Searching is the act of checking for and retrieving a specific data element, whereas sorting rearranges multiple data elements in an array or list into some kind of order.

What Is the Difference Between a Static and a Dynamic Programming Language?

Source: [DZone](http://dzone.com/)

Static language:

* The coder must declare the data types of variables before they can be assigned a value.
* The computer will check these data types when the code is compiled, and any errors must be fixed before the code can be run.

Dynamic language:

* Data types do not need to be declared by the coder.
* The data types will be checked at runtime, meaning that code containing errors will attempt to run.