

## 2. Depth-first search

Depth-first search (or DFS) is a traversal option for trees or graphs in which child nodes are visited before their siblings on the same layer. [Learn more](#).

### 2.1 Easy depth-first search questions

#### Question 1: Binary tree inorder traversal

- [Text guide](#) (Dev.to/javinpaul)
- [Video guide](#) (DEEPTI TALESRA)
- [Code example](#) (OldCodingFarmer)

#### Question 2: Same tree

- [Text guide](#) (Fatal Errors)
- [Text guide](#) (LeetCode)
- [Video guide](#) (NeetCode)
- [Code example](#) (OldCodingFarmer)

#### Question 3: Symmetric tree

- [Text guide](#) (Baeldung)
- [Text guide](#) (Medium/Hary Krishnan)
- [Video guide](#) (Back To Back SWE)
- [Code example](#) (lvllolitte)

### 2.2 Medium depth-first search questions

#### Question 4: Surrounded regions

- [Text guide](#) (After Academy)
- [Video guide](#) (NeetCode)
- [Video guide](#) (Nick White)
- [Code example](#) (Chronoviser)

#### Question 5: Clone graph

- [Text guide](#) (Medium/Deeksha Sharma)
- [Video guide](#) (Michael Muinos)
- [Video guide](#) (NeetCode)
- [Code example](#) (mohamede1945)

#### Question 6: Number of islands

- [Text guide](#) (Codertrain)

- [Video guide](#) (Kevin Naughton Jr.)
- [Code example](#) (girikuncoro)

#### Question 7: Course schedule

- [Text guide](#) (Yu's Coding)
- [Video guide](#) (NeetCode)
- [Video guide](#) (Nideesh Terapalli)
- [Code example](#) (varunu28)

## 2.3 Hard depth-first search questions

#### Question 8: Binary tree maximum path sum

- [Text guide](#) (After Academy)
- [Video guide](#) (NeetCode)
- [Video guide](#) (Michael Muinos)
- [Code example](#) (arkaung)

#### Question 9: Alien dictionary

- [Text guide](#) (Medium/Feng Wang)
- [Video guide](#) (NeetCode)
- [Video guide](#) (happygirlzt)
- [Code example](#) (Tenderleo)

#### Question 10: Serialize and deserialize binary tree

- [Text guide](#) (Baeldung)
- [Video guide](#) (Back To Back SWE)
- [Code example](#) (gavinlinasd)

How did you get on? To practice with more depth-first search questions like this, check out our list of [50+ depth-first search questions with solutions](#).

## 3. Breadth-first search

Breadth-first search (or BFS) is one traversal method for trees and graphs in which all vertices on one layer are visited before visiting their children on the next layer – i.e. every node on layer  $i$  is visited before the nodes on layer  $i+1$ . [Learn more](#).

### 3.1 Easy breadth-first search questions

#### Question 11: Maximum depth of binary tree

- [Text guide](#) (Aaronice)

- [Video guide](#) (NeetCode)
- [Code example](#) (makuiyu)

#### Question 12: Minimum depth of binary tree

- [Text guide](#) (After Academy)
- [Video guide](#) (Terrible Whiteboard)
- [Code example](#) (simkieu)

#### Question 13: Maximum depth of N-ary tree

- [Text guide](#) (Medium/Annamariya Tharayil)
- [Video guide](#) (Nick White)
- [Code example](#) (TKroos)

### 3.2 Medium breadth-first search questions

#### Question 14: Binary tree level order traversal

- [Text guide](#) (Educative)
- [Video guide](#) (Kevin Naughton Jr.)
- [Video guide](#) (Nick White)
- [Code example](#) (SOY)

#### Question 15: Binary tree zigzag level order traversal

- [Text guide](#) (zhenyu0519)
- [Video guide](#) (Michael Vandi)
- [Code example](#) (fangkunjsy)

#### Question 16: Binary tree level order traversal II

- [Text guide](#) (Medium/Navneet Ojha)
- [Video guide](#) (Terrible Whiteboard)
- [Code example](#) (OldCodingFarmer)

### 3.3 Hard breadth-first search questions

#### Question 17: Word ladder

- [Text guide](#) (Medium/Nathan Brickett)
- [Video guide](#) (Nick White)
- [Code example](#) (jianchao-li)

### Question 18: Word ladder II

- [Text guide](#) (medium/spylogsster)
- [Video guide](#) (TECH DOSE)
- [Code example](#) (antdavid)

### Question 19: Cut off trees for golf event

- [Text guide](#) (Zirui)
- [Video guide](#) (Shivam Patel)
- [Code example](#) (shawngao)

Keen to work through some more? See our article, [44 breadth-first search questions and solutions](#). That should keep you busy for a while!

## 4. Binary search

Binary search is one of the fastest search algorithms because it halves the search space with each iteration. It requires an ordered set that also has constant access times, meaning that only sorted arrays are suitable for binary search. [Learn more](#).

### 4.1 Easy binary search questions

#### Question 20: Search insert position

- [Text guide](#) (GeeksForGeeks)
- [Video guide](#) (NeetCode)
- [Video guide](#) (TECH DOSE)

#### Question 21: Sqrt(x)

- [Text guide](#) (GoodTecher)
- [Video guide](#) (Terrible Whiteboard)
- [Code example](#) (AlexTheGreat)

#### Question 22: First bad version

- [Text guide](#) (Medium/Mac Sampson)
- [Video guide](#) (Kevin Naughton Jr.)
- [Code example](#) (Cheng\_Zhang)

### 4.2 Medium binary search questions

#### Question 23: Search in a rotated sorted array

- [Text guide](#) (Educative)

- [Video guide](#) (NeetCode)
- [Video guide](#) (Errichto)
- [Code example](#) (1337beef)

#### **Question 24: Find first and last position of element in sorted array**

- [Text guide](#) (Enjoy Algorithms)
- [Video guide](#) (Nick White)
- [Code example](#) (stellari)

#### **Question 25: Search a 2D matrix**

- [Text guide](#) (TheLeanProgrammer)
- [Video guide](#) (NeetCode)
- [Code example](#) (LeetCode)

### **4.3 Hard binary search questions**

#### **Question 26: Median of two sorted arrays**

- [Text guide](#) (Medium/hamid)
- [Video guide](#) (NeetCode)
- [Code example](#) (MissMary)

#### **Question 27: Count of smaller numbers after self**

- [Text guide](#) (Seanforfun)
- [Video guide](#) (happygirlzt)
- [Code example](#) (mayanist)

#### **Question 28: Find minimum in rotated sorted array II**

- [Text guide](#) (GraceMeng)
- [Video guide](#) (Naresh Gupta)

For more questions like these, check out our list of [50 binary search questions and solutions](#).

## **5. Sorting**

A sorting algorithm can be performed on arrays or linked lists, in order to rearrange elements according to a series of instructions. Many algorithms require, or perform better on, a sorted dataset. [Learn more](#).

### **5.1 Easy sorting questions**

### Question 29: Contains duplicate

- [Text guide](#) (Medium/punitkmryh)
- [Video guide](#) (Nick White)
- [Code example](#) (jmnarloch)

### Question 30: Valid anagram

- [Text guide](#) (Project Debug)
- [Video guide](#) (Nick White)
- [Code example](#) (OldCodingFarmer)

### Question 31: Meeting rooms

- [Text guide](#) (Aaronice)
- [Video guide](#) (NeetCode)
- [Code example](#) (Seanforfun)

## 5.2 Medium sorting questions

### Question 32: Sort an array

- [Text guide](#) (cc189)
- [Video guide](#) (leettuition)
- [Code example](#) (HaelChan)

### Question 33: 3Sum

- [Text guide](#) (fizzbuzzed)
- [Video guide](#) (Nick White)
- [Video guide](#) (NeetCode)
- [Code example](#) (shpolsky)

### Question 34: H-index

- [Text guide](#) (TitanWolf)
- [Video guide](#) (Algorithms Made Easy)
- [Code example](#) (yfcheng)

## 5.3 Hard sorting questions

### Question 35: Maximum gap

- [Text guide](#) (Buttercola)
- [Video guide](#) (Coding Decoded)
- [Code example](#) (zkfairytale)

### Question 36: Merge k sorted lists

- [Text guide](#) (After Academy)
- [Video guide](#) (NeetCode)
- [Video guide](#) (Kevin Naughton Jr.)
- [Code example](#) (reeclapple)

### Question 37: Count of smaller numbers after self

- [Text guide](#) (Mithlesh Kumar)
- [Video guide](#) (happygirlzt)
- [Code example](#) (confiscate)

Need some more sorting questions to practice with? No problem, we've got plenty more here: [54 sorting questions and solutions.](#)

## 6. Dynamic programming

Dynamic programming is an algorithmic paradigm used to create optimal solutions for complex problems by breaking them down into simpler sub-problems that can be solved recursively. [Learn more.](#)

### 6.1 Easy dynamic programming questions

#### Question 38: Maximum subarray

- [Text guide](#) (Baeldung)
- [Video guide](#) (Back to Back SWE)
- [Code example](#) (FujiwaranoSai)

#### Question 39: Climbing stairs

- [Text guide](#) (Medium/Analytics Vidhya)
- [Video guide](#) (NeetCode)
- [Code example](#) (liaison)

#### Question 40: Pascal's triangle

- [Text guide](#) (Dev.to/seanpgallivan)
- [Video guide](#) (Nick White)
- [Video guide](#) (NeetCode)
- [Code example](#) (rheaxu)

### 6.2 Medium dynamic programming questions

#### Question 41: Longest palindromic substring

- [Text guide](#) (RedQuark)
- [Video guide](#) (NeetCode)
- [Video guide](#) (Errichto)

#### Question 42: Unique paths

- [Text guide](#) (Medium/Arpit Choudhary)
- [Video guide](#) (NeetCode)
- [Video guide](#) (Kevin Naughton Jr.)
- [Code example](#) (jianchao-li)

#### Question 43: Unique paths II

- [Text guide](#) (Medium/Nerd For Tech)
- [Video guide](#) (TECH DOSE)
- [Code example](#) (tusizi)

### 6.3 Hard dynamic programming questions

#### Question 44: Regular expression matching

- [Text guide](#) (RedQuark)
- [Video guide](#) (Tushar Roy)
- [Video guide](#) (NeetCode)
- [Code example](#) (LeetCode)

#### Question 45: Maximal rectangle

- [Text guide](#) (Rohith Vazhathody)
- [Video guide](#) (Knapsak)
- [Video guide](#) (TECH DOSE)
- [Code example](#) (morrishen2008)

#### Question 46: Edit distance

- [Text guide](#) (After Academy)
- [Video guide](#) (Back to Back SWE)
- [Video guide](#) (Tushar Roy)
- [Code example](#) (anderson5)

Check out plenty more dynamic programming questions here: [53 dynamic programming questions and solutions.](#)



## 7. Greedy algorithms

A greedy algorithm is an algorithmic paradigm that finds the optimal solution to a problem by breaking the problem down into smaller (local) parts and finding the best solution for each of these parts. [Learn more](#).

### 7.1 Easy greedy algorithm questions

#### Question 47: Assign cookies

- [Text guide](#) (Medium/Fatboy Slim)
- [Video guide](#) (Nick White)
- [Video guide](#) (Kevin Naughton Jr.)
- [Code example](#) (fabrizio)

#### Question 48: Can place flowers

- [Text guide](#) (dilyar85)
- [Video guide](#) (NeetCode)
- [Video guide](#) (Nideesh Terapalli)
- [Code example](#) (soumyadeep2007)

#### Question 49: Lemonade change

- [Text guide](#) (Tutorial Cup)
- [Video guide](#) (Nick White)
- [Code example](#) (lee215)

### 7.2 Medium greedy algorithm questions

#### Question 50: Jump game

- [Text guide](#) (Learnbay)
- [Video guide](#) (NeetCode)
- [Code example](#) (1337beef)

#### Question 51: Gas station

- [Text guide](#) (After Academy)
- [Video guide](#) (NeetCode)
- [Code example](#) (daxianji007)

#### Question 52: Jump game II

- [Text guide](#) (Medium/Nerd For Tech)
- [Video guide](#) (NeetCode)

- [Code example](#) (Cheng Zhang)

## 7.3 Hard greedy algorithm questions

### Question 53: Candy

- [Text guide](#) (LeetCode)
- [Video guide](#) (Algorithms Made Easy)
- [Code example](#) (meng789987)

### Question 54: Create maximum number

- [Text guide](#) (Medium/dume0011)
- [Code example](#) (dietpepsi)

### Question 55: Patching array

- [Text guide](#) (HackingNote)
- [Video guide](#) (Timothy H Chang)
- [Code example](#) (StefanPochmann)

For plenty more greedy algorithm questions, see [50 greedy algorithm interview questions](#).

## 8. Backtracking

Backtracking is a form of brute-force problem solving, but with the ability to discard potential solutions early, before they are fully explored. It is an algorithmic paradigm for incrementally finding solutions to problems. [Learn more.](#)

### 8.1 Easy backtracking questions

Backtracking questions don't tend to be very easy, so we've only included one example in this first section.

### Question 56: Binary watch

- [Text guide](#) (Medium/Competitive Programming)
- [Video guide](#) (Owen Smith)
- [Code example](#) (xietao0221)

### 8.2 Medium backtracking questions

### Question 57: Letter combinations of a phone number

- [Text guide](#) (After Academy)
- [Video guide](#) (NeetCode)

- [Video guide](#) (Kevin Naughton Jr.)
- [Code example](#) (lei31)

#### Question 58: Generate parentheses

- [Text guide](#) (RedQuark)
- [Video guide](#) (Back to Back SWE)
- [Code example](#) (brobins9)

#### Question 59: Permutations

- [Text guide](#) (issac3)
- [Video guide](#) (NeetCode)
- [Code example](#) (OldCodingFarmer)

### 8.3 Hard backtracking questions

#### Question 60: Word break II

- [Text guide](#) (Java questions)
- [Video guide](#) (babybear4812)
- [Code example](#) (Cheng\_Zhang)

#### Question 61: Sudoku solver

- [Text guide](#) (After Academy)
- [Video guide](#) (Back To Back SWE)
- [Code example](#) (LeetCode)

#### Question 62: Stickers to spell word

- [Text guide](#) (linlaw)
- [Video guide](#) (NeetCode)
- [Code example](#) (zestypanada)

You can find lots more backtracking questions to practice with right here: [47 backtracking interview questions](#).

## 9. Divide and conquer

Divide and conquer is an algorithmic paradigm used to solve problems by continually dividing the problem into smaller parts until a part is easy enough to solve (conquer) on its own. The solutions to the solved parts are then combined to give the solution for the original problem. [Learn more](#).

### 9.1 Easy divide and conquer questions

### Question 63: Maximum subarray

- [Text guide](#) (Techie Delight)
- [Video guide](#) (Ghassan Shobaki)
- [Code example](#) (jianchao-li)

### Question 64: Majority element

- [Text guide](#) (enjoy algorithms)
- [Video guide](#) (Nideesh Terapalli)
- [Code example](#) (coderoath)

### Question 65: First bad version

- [Text guide](#) (Studytonight)
- [Video guide](#) (TECH DOSE)
- [Video guide](#) (Kevin Naughton Jr.)
- [Code example](#) (RainbowSecret)

## 9.2 Medium divide and conquer questions

### Question 66: Kth largest element in an array

- [Text guide](#) (Coder's Cat)
- [Video guide](#) (NeetCode)
- [Code example](#) (jmnarloch)

### Question 67: Search a 2D matrix II

- [Text guide](#) (Medium/Nerd for tech)
- [Video guide](#) (Back to Back SWE)
- [Code example](#) (LeetCode)

### Question 68: Longest substring with at least K repeating characters

- [Text guide](#) (Medium/dume0011)
- [Video guide](#) (Knowledge Center)
- [Code example](#) (cdpiano)

## 9.3 Hard divide and conquer questions

### Question 69: Median of two sorted arrays

- [Text guide](#) (Medium/hamid)
- [Video guide](#) (NeetCode)
- [Code example](#) (MissMary)

### Question 70: Reverse pairs

- [Text guide](#) (Tutorialhorizon)
- [Video guide](#) (take U forward)
- [Code example](#) (fun4LeetCode)

### Question 71: Count of smaller numbers after self

- [Text guide](#) (Programmerall)
- [Video guide](#) (happygirlzt)
- [Code example](#) (confiscate)

For lots more divide and conquer questions, see our article [50 divide and conquer interview questions and cheat sheet](#).