

---

[Coaches](#) [Peers](#) [Courses](#) [Articles](#) [Stories](#)

---

# 50+ array questions with solutions (easy, medium, hard)

SOFTWARE ENGINEERING

AUG. 13, 2021

---

## Table of contents

[Easy](#)[Medium](#)[Hard](#)[Basics](#)[Cheat sheet](#)[Mocks](#)[Practice with ex-FAANG interviewers](#)

---

number.

- Given a 2D array of integers, rotate clockwise without using additional memory.
- If you have two sorted arrays, how can you merge them and keep the resulting array sorted?
- Given unlimited coins in denominations of 1c, 2c, and 5c, how many different ways can you make a total of 20c? Can you solve the general version of this problem for an arbitrary target amount and a given list of denominations?

Below, we take a look at some more questions and provide you with links to high quality solutions to them. We explain how arrays work, their variations, and the most important things you need to know about them, including a useful 'cheat sheet' to remind you of the key points at a glance.

This is an overview of what we'll cover:

1. [Easy array interview questions](#)
2. [Medium array interview questions](#)
3. [Hard array interview questions](#)
4. [Array basics](#)
5. [Array cheat sheet](#)

---

Table of contents

[Easy](#)

[Medium](#)

[Hard](#)

[Basics](#)

[Cheat sheet](#)

[Mocks](#)

[Practice with ex-FAANG interviewers](#)

Here are some of the easiest questions you might get asked in a coding interview. These questions are often asked during the 'phone screen' stage, so you should be comfortable answering them without being able to write code or use a whiteboard.

#### 1.1 Merge two sorted arrays

- [Text guide](#) (GeeksforGeeks)
- [Video guide](#) (TECH DOSE)

#### 1.2 Remove duplicates from an array

- [Video guide](#) (Kevin Naughton Jr.)
- [Text guide](#) (W3Schools)
- [Text guide](#) (javarevisted)
- [Code example](#) (LeetCode)

#### 1.3 Count the frequency of an element in an array

- [Text guide](#) (GeeksforGeeks)
- [Video guide](#) (SDET)

---

## Table of contents

[Easy](#)

[Medium](#)

[Hard](#)

[Basics](#)

[Cheat sheet](#)

[Mocks](#)

[Practice with ex-FAANG interviewers](#)

- [Video guide](#) (Kevin Naughton Jr.)
- [Video guide](#) (Take u Forward)

#### 1.7 Remove element in-place

- [Text guide](#) (Redquark)
- [Video guide](#) (Nick White)
- [Code example](#) (LeetCode)

#### 1.8 Search Insert Position

- [Text guide](#) (GeeksforGeeks)
- [Text guide](#) (Codesdope)
- [Video guide](#) (NeetCode)

#### 1.9 Maximum Subarray

- [Text guide](#) (Wikipedia)
- [Text guide](#) (Techie Delight)

[Video guide](#) (CS Do IT)

---

## Table of contents

### [Easy](#)

### [Medium](#)

### [Hard](#)

### [Basics](#)

### [Cheat sheet](#)

### [Mocks](#)

- [Video guide](#) (Kevin Naughton Jr)
- [Video guide](#) (TECH DOSE)

#### 1.13 Count Primes

- [Text guide](#) (GeeksForGeeks)
- [Video guide](#) (Terrible Whiteboard)

#### 1.14 Contains Duplicate

- [Text guide](#) (Medium/Punitkmryh)
- [Video guide](#) (Terrible Whiteboard)
- [Code example](#) (LeetCode)

#### 1.15 Third Largest Number

- [Text guide](#) (GeeksForGeeks)
- [Video guide](#) (Terrible Whiteboard)

#### 1.16 Count Odd Even

---

## Table of contents

### [Easy](#)

### [Medium](#)

### [Hard](#)

### [Basics](#)

### [Cheat sheet](#)

### [Mocks](#)

---

## Text guide (LeetCode)

- [Video guide](#) (Programming tutorials)
- [Code example](#) (LeetCode)

### 2.2 Find if a given element is in a sorted array (binary search)

- [Text guide](#) (Khan academy)
- [Video guide](#) (HackerRank)
- [Code example](#) (LeetCode)

### 2.3 Rotate an array

- [Text guide](#) (GeeksforGeeks)
- [Video guide](#) (Nick White)
- [Code example](#) (LeetCode)

### 2.4 Largest sum of non-adjacent numbers (Dynamic Programming)

- [Text guide](#) (Medium/Arun Kumar)
- [Text guide](#) (GeeksForGeeks)

---

## Table of contents

### Easy

### Medium

### Hard

### Basics

### Cheat sheet

### Mocks

---

## 2.7 Shortest Unsorted Continuous Subarray

- [Text guide](#) (GeeksForGeeks)
- [Text guide](#) (Seanpgallivan)
- [Video guide](#) (Nick White)

## 2.8 Maximum sum of hour glass in matrix

- [Text guide](#) (GeeksforGeeks)
- [Video guide](#)(Over The Shoulder Coding)

## 2.9 Paint House (Dynamic programming)

- [Text guide](#) (ProgrammerSought)
- [Video guide](#) (NeetCode)

## 2.10 Minimum number of jumps to reach end

- [Text guide](#) (Medium/Himanshu)
- [Video guide](#) (Nick White)

---

## Table of contents

[Easy](#)

[Medium](#)

[Hard](#)

[Basics](#)

[Cheat sheet](#)

[Mocks](#)

[Practice with ex-FAANG interviewers](#)

#### 2.14 Minimum number of swaps to sort an array

- [Text guide](#) (GeeksForGeeks)
- [Video guide](#) (Brian Dyck)

## 3. Hard array interview questions

Similar to the moderate section, these more difficult questions may be asked in an onsite or video call interview. You will likely be given more time if you are expected to create a full solution.

#### 3.1 Rotate a 2D array

- [Text guide](#) (Jack)
- [Text guide](#) (GeeksforGeeks)
- [Video guide](#) (Nick White)

#### 3.2 Create change with coins (dynamic programming)

---

### Table of contents

#### [Easy](#)

#### [Medium](#)

#### [Hard](#)

#### [Basics](#)

#### [Cheat sheet](#)

#### [Mocks](#)



3.6 Find the maximum  $j - i$  such that  $arr[j] > arr[i]$  (Distance maximising problem)

- [Text guide](#) (GeeksForGeeks)
- [Text guide](#) (Techie Delight)
- [Video guide](#) (Genetic Coders)

3.7 Array manipulation

- [Text guide](#) (The Poor Coder)
- [Video guide](#) (Brian Dyck)

3.8 Median of Two Sorted Arrays

- [Text guide](#) (GeeksForGeeks)
- [Video guide](#) (Take u Forward)
- [Code example](#) (LeetCode)

3.9 Sudoku Solver

---

## Table of contents

[Easy](#)

[Medium](#)

[Hard](#)

[Basics](#)

[Cheat sheet](#)

[Mocks](#)

[Practice with ex-FAANG interviewers](#)

---

### 3.12 Find Minimum in Rotated Sorted Array

- [Text guide](#) (Algorithmsandme)
- [Video guide](#) (Nick White)

### 3.13 Count of Smaller Numbers After Self

- [Text guide](#) (CodeStudio)
- [Text guide](#) (GeeksForGeeks)
- [Video guide](#) (Happygirlzt)

### 3.14 Palindrome Pairs

- [Text guide](#) (Seanpgallivan)
- [Video guide](#) (Happygirlzt)

### 3.15 Sort an array containing 0's, 1's and 2's

- [Text guide](#) (Techie Delight)
- [Text guide](#) (GeeksForGeeks)
- [Video guide](#) (Take u Forward)

---

## Table of contents

[Easy](#)

[Medium](#)

[Hard](#)

[Basics](#)

[Cheat sheet](#)

[Mocks](#)

[Practice with ex-FAANG interviewers](#)

## 4. Array basics

In order to crack the questions above and others like them, you'll need to have a strong understanding of arrays, how they work, and when to use them. Let's get into it.

### 4.1 What is an array?

An array is a list-like data structure that contains a collection of values, each associated with a specific index, usually with a fixed overall size. For example, the image below shows an array that has space for up to nine elements, but contains only four. This array has the integers 1, 2, 3, and 4 as its values and these are at the “zeroth”, first, second, and third indices respectively.

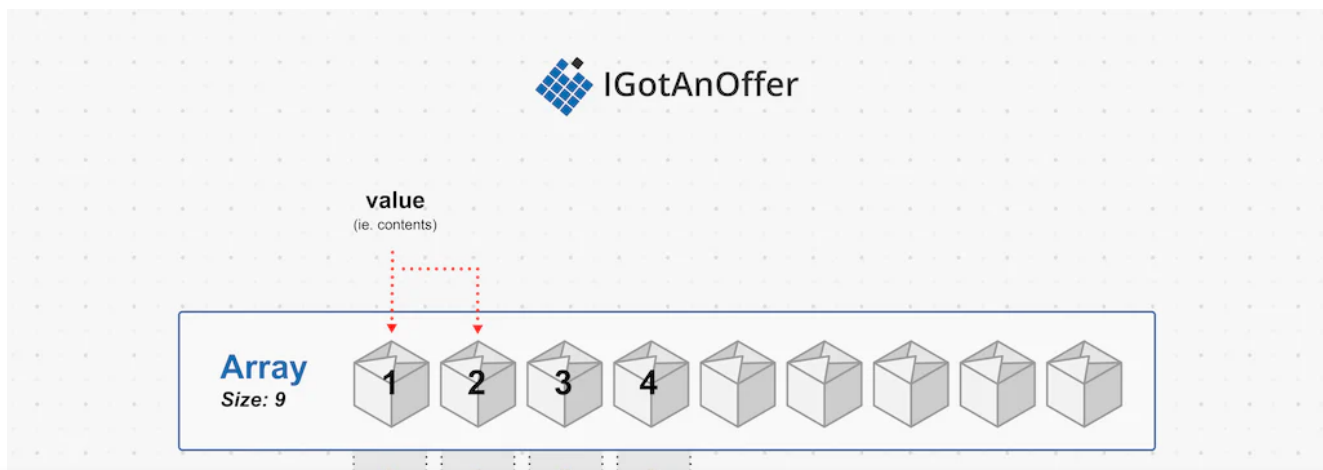


Table of contents

[Easy](#)

[Medium](#)

[Hard](#)

[Basics](#)

[Cheat sheet](#)

[Mocks](#)

[Practice with ex-FAANG interviewers](#)

Java and C++), and usually offer alternative implementations that the user can import from a standard library.

In many languages, including Java, default arrays are static and homogenous. Static means that the size of the array (the number of elements that it can hold) has to be declared upfront, when the array is created. Homogenous means that all of the elements in the array must be of the same type - e.g. an array of integers cannot contain string or float elements.

In other languages, including Python, the default array (`list`) is dynamic and heterogeneous. This means that they can be resized dynamically at run time, and can contain a mix of different types.

You will also often encounter nested or multidimensional arrays (often called a matrix). For 2D arrays, you can usually think of these as tables with rows and columns.

Because array terminology and implementation differs across languages, it's always a good idea to check your assumptions about a specific array question with your interviewer.

## 4.1.2 How arrays store data

As with strings, data stored in arrays is traditionally kept in the heap of computer memory. If you store a basic integer in a variable with a statement like `int x = 1;`, that value is stored on the stack. To answer many array-related interview questions, you

Table of contents

Easy

Medium

Hard

Basics

Cheat sheet

Mocks

A common approach that languages use for dynamic arrays is to double their allocated size every time they become full. So if you need to add an 11th item to an array of size 10, the library will create a new array of size 20 and copy across the existing data.

This means that as you are adding elements to an array, most inserts will be fast, but your code will slow down significantly every time it triggers a resize.

### 4.1.3 How arrays compare to other data structures

Because strings are usually implemented as arrays of characters, many interview questions for arrays can be phrased as string interview questions, and vice-versa.

Arrays are also closely related to linked lists, and many questions will expect you to be able to explain the differences between them, and when one has an advantage over the other.

Finally, arrays are often contrasted with sets. When you want to get data at a specific index (e.g. "I need the fifth element in this list"), arrays perform better than sets, as you can access any given element by its index in  $O(1)$  time.

If you need to check if a specific value is contained in the array ("Does my array contain the value 5 at any position?"), arrays are not efficient. You need to loop through every single value to see if it matches what you are looking for, while sets can provide this in  $O(1)$  time.

---

[Table of contents](#)

[Easy](#)

[Medium](#)

[Hard](#)

[Basics](#)

[Cheat sheet](#)

[Mocks](#)

[Practice with ex-FAANG interviewers](#)

# Array Cheat Sheet

(Space-Time complexity)

## Time complexity:

	Worst Case Scenario	Average Case Scenario	Best Case Scenario
Accessing an element	$O(1)$	$O(1)$	$O(1)$
Updating an element	$O(1)$	$O(1)$	$O(1)$
Deleting an element	$O(n)$	$O(n)$	$O(1)$
Inserting an element	$O(n)$	$O(n)$	$O(1)$
Searching for an element	$O(n)$	$O(n)$	$O(1)$

## Complexity of algorithms:

	Time Complexity			Space Complexity
	Worst Case	Average Case	Best Case	
Quicksort	$O(n^2)$	$O(n \log(n))$	$O(n \log(n))$	$O(\log(n))$
Mergesort	$O(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$	$O(n)$
Heapsort	$O(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$	$O(1)$
Bubble Sort	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$
Insertion Sort	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$
Selection Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$
Binary Search	$O(\log(n))$	$O(\log(n))$	$O(1)$	$O(1)$
Linear Search	$O(n)$	$O(n)$	$O(1)$	$O(1)$

Table of contents

Easy

Medium

Hard

Basics

Cheat sheet

Mocks

Practice with ex-FAANG interviewers

- [Looping](#)
- [Sorting](#)

## 5.2 Related concepts

- Homogeneous (elements have same type)
- Dynamic (size can change)

## 5.3 Cheat sheet explained

The cheat sheet above is a summary of information you might need to know for an interview, but it's usually not enough to simply memorize it. Instead, aim to understand each result so that you can give the answer in context.

The cheat sheet is broken into time complexity (the processing time for the various array operations) and space complexity (the amount of memory required). While you might be asked about these directly in relation to the array data structure, it's more likely that you will need to know these in relation to specific array-related algorithms, such as searching and sorting, which is what the third section details.

For more information about time and space requirements of different algorithms, read our [complete guide to big-O notation and complexity analysis](#).

---

### Table of contents

---

#### [Easy](#)

---

#### [Medium](#)

---

#### [Hard](#)

---

#### [Basics](#)

---

#### [Cheat sheet](#)

---

#### [Mocks](#)

---

our best case is  $O(1)$ .

### 5.3.2 Space complexity

In most cases, the space complexity of an array is simply the number of elements, so this is  $O(n)$ . In some contexts, the array might be some (small) constant size, which means the space complexity is simplified to  $O(1)$ . Space complexity is almost always only relevant in the context of a specific algorithm, which we cover in the next section.

### 5.3.3 Array algorithms complexity

We've listed the algorithms that interviewers will most frequently discuss while asking about arrays, but there are dozens of other search algorithms and sorting algorithms. One of the most important aspects to understand is the tradeoff between mergesort and quicksort. Quicksort works in place, so does not require additional memory, while Mergesort uses an auxiliary array, and therefore uses more space. On the flip side, the worst time complexity of mergesort is better than that of quicksort which can in some cases (e.g. when the array is already sorted) perform as badly as a naive bubble sort.

For the search algorithms, a key insight to understand is that binary search is  $\log(n)$  as we can eliminate half of the array with each operation. Therefore doubling the size of the array requires only one more operation. By contrast, a linear search looks at every element until it finds the target, so doubling the size of the array also requires, on average, twice as many operations.

---

Table of contents

Easy

Medium

Hard

Basics

Cheat sheet

Mocks



- [Trees](#)
- [Graphs](#)
- [Maps](#)
- [Heap](#)
- [73 data structure interview questions](#) (with solutions and cheat sheet)

Once you're confident on all the topics, you'll want to start practicing answering coding questions in an interview situation.

One way of doing this is by practicing out loud, which is a very underrated way of preparing. However, sooner or later you're probably going to want some expert interventions and feedback to really improve your interview skills.

That's why we recommend practicing with ex-interviewers from top tech companies. If you know a software engineer who has experience running interviews at a big tech company, then that's fantastic. But for most of us, it's tough to find the right connections to make this happen. And it might also be difficult to practice multiple hours with that person unless you know them really well.

Here's the good news. We've already made the connections for you. We've created a coaching service where you can practice system design interviews 1-on-1 with ex-interviewers from leading tech companies. [Learn more and start scheduling sessions today.](#)

---

## Table of contents

### [Easy](#)

### [Medium](#)

### [Hard](#)

### [Basics](#)

### [Cheat sheet](#)

### [Mocks](#)

[Practice with ex-FAANG interviewers](#)

# Any questions about array interview questions?

If you have any questions about arrays or coding interviews in general, don't hesitate to ask them in the comments below. All questions are good questions, so go ahead!



---

## Table of contents

[Easy](#)[Medium](#)[Hard](#)[Basics](#)[Cheat sheet](#)[Mocks](#)[Practice with ex-FAANG interviewers](#)

---

## COMPANY

[Become a coach](#)[About](#)[Privacy](#)

## INTERVIEW GUIDES

[Facebook PM interview](#)[Google PM interview](#)[Amazon PM interview](#)

---

[Table of contents](#)

---

[Easy](#)[Medium](#)

---

[Hard](#)[Basics](#)[Cheat sheet](#)[Mocks](#)[Practice with ex-FAANG interviewers](#)

Copyright © 2022, IGotAnOffer.

---

## Table of contents

---

[Easy](#)

[Medium](#)

[Hard](#)

[Basics](#)

[Cheat sheet](#)

[Mocks](#)

[Practice with ex-FAANG interviewers](#)