

# DsAlgo.com

## Index

- [Preface](#)
- [Categorized index](#)
- 1. [Find maximum and minimum](#)
- 2. [Make larger number](#)
- 3. [Next larger palindrome](#)
- 4. [Least difference in array](#)
- 5. [Print matrix spiral](#)
- 6. [Move zeros to the right](#)
- 7. [Find repetition multiple sorted arrays](#)
- 8. [Largest sum sub array](#)
- 9. [Search in a sorted matrix](#)
- 10. [Kth largest in sorted matrix](#)
- 11. [Largest palindrome iterative](#)
- 12. [Reverse words of sentence](#)

We are presenting a collection of data structure and algorithm questions and answers for technical interviews for software companies. Questions are collected from real interviews of companies like Microsoft, Amazon, Facebook, Google or Yahoo. Questions are solved and the solutions are discussed in an optimal way which enables you to do a quick preparation for interview. A complete running Java program is added with each problem which you can copy and run in your IDE to understand the solution in a better way. Your feedback is very much appreciated. Please help the site grow by sharing your own problem and solutions or suggestions. Please contact us through [feedback@dsalgo.com](mailto:feedback@dsalgo.com)

## Categorized content

13. [Rotate array k times](#)
14. [Merge in single array](#)
15. [Rotate a string to make another](#)
16. [Maximum product subarray](#)
17. [Maximum sum sub matrix](#)
18. [Expand the array](#)
19. [Sort to bring anagrams closer](#)
20. [Two missing numbers](#)
21. [Maximize stock profit simple](#)
22. [Sum of array except current element](#)
23. [Maximum arithmetic sequence](#)
24. [Reverse linked list iterative](#)
25. [Reverse linked list recursive](#)
26. [Fold a linked list](#)
27. [Reverse k nodes in linked list](#)
28. [Find loop in linked list](#)
29. [Linked list Y shape](#)
30. [Kth node from end](#)
31. [Get find delete in O\(1\)](#)
32. [Calculate power](#)

## Array

[Find maximum and minimum](#)  
[Make larger number](#)  
[Next larger palindrome](#)  
[Least difference in array](#)  
[Print matrix spiral](#)  
[Move zeros to the right](#)  
[Find repetition multiple sorted arrays](#)  
[Largest sum sub array](#)  
[Search in a sorted matrix](#)  
[Kth largest in sorted matrix](#)  
[Largest palindrome iterative](#)  
[Reverse words of sentence](#)  
[Rotate array k times](#)  
[Merge in single array](#)  
[Rotate a string to make another](#)  
[Maximum product subarray](#)  
[Maximum sum sub matrix](#)  
[Expand the array](#)  
[Sort to bring anagrams closer](#)  
[Two missing numbers](#)  
[Maximize stock profit simple](#)  
[Sum of array except current element](#)  
[Maximum arithmetic sequence](#)

## Linked list

33. [Next power of two](#)
34. [Total 1s in numbers till n](#)
35. [Swap without temp](#)
36. [Array next element](#)
37. [Stack with get minimum](#)
38. [Sort a stack](#)
39. [Find all permutations](#)
40. [Find all paths in a maze](#)
41. [Find longest path in maze](#)
42. [Towers of Hanoi](#)
43. [Snakes and ladders](#)
44. [Queue using stack](#)
45. [Queue minimum using stack](#)
46. [Shortest path in a maze](#)
47. [All unique letter substring](#)
48. [Linked list remove duplicate](#)
49. [Are two words anagram](#)
50. [Longest subarray equal 1 & 0](#)
51. [Pythagorean triples](#)
52. [Linked list with random pointer](#)
53. [Same average subset](#)
54. [Anagram substring search](#)
55. [Balance the balance](#)

[Reverse linked list iterative](#)  
[Reverse linked list recursive](#)  
[Fold a linked list](#)  
[Reverse k nodes in linked list](#)  
[Find loop in linked list](#)  
[Linked list Y shape](#)  
[Kth node from end](#)  
[Get find delete in O\(1\)](#)  
Bits[Calculate power](#)  
[Next power of two](#)  
[Total 1s in numbers till n](#)  
[Swap without temp](#)

## Stack

[Array next element](#)  
[Stack with get minimum](#)  
[Sort a stack](#)  
[Find all permutations](#)  
[Find all paths in a maze](#)  
[Find longest path in maze](#)  
[Towers of Hanoi](#)

## Queue

[Snakes and ladders](#)  
[Queue using stack](#)  
[Queue minimum using stack](#)

- 56. [Lowest common ancestor](#)
- 57. [Sum of child nodes](#)
- 58. [Lowest common ancestor without root](#)
- 59. [Binary tree zigzag print](#)
- 60. [Print Binary tree bottom to top](#)
- 61. [Binary tree bottom to top level wise](#)
- 62. [Print Nodes of the same level](#)
- 63. [Linked list with inorder successor](#)
- 64. [Linked list with preorder successor](#)
- 65. [Linked list with postorder successor](#)
- 66. [Binary tree to linked list](#)
- 67. [Is the binary tree BST](#)
- 68. [Create tree from in and pre](#)
- 69. [Find root to node path](#)
- 70. [Find distance between two nodes](#)
- 71. [SuperImpose Binary Tree](#)
- 72. [Is a sum possible along any path](#)
- 73. [Remove duplicate infinite](#)

[Shortest path in a maze](#)

## Hash

[All unique letter substring](#)

[Linked list remove duplicate](#)

[Are two words anagram](#)

[Longest subarray equal 1 & 0](#)

[Pythagorean triples](#)

## Map

[Linked list with random pointer](#)

[Same average subset](#)

[Anagram substring search](#)

## Heap

[Running Median](#)

[Maximum k integers using min heap](#)

[Max heap and BST in one](#)

[Merge N sorted arrays](#)

## Backtracking

[Separate words in sentence](#)

[Increasing decreasing tuple](#)

## Tree

[Balance the balance](#)

[Lowest common ancestor](#)

[Sum of child nodes](#)

73. <a href="#">Remove duplicate infinite integer</a>	<a href="#">Lowest common ancestor without root</a>
74. <a href="#">Find deepest level nodes</a>	<a href="#">Binary tree zigzag print</a>
75. <a href="#">Maximum sum path positive</a>	<a href="#">Print Binary tree bottom to top</a>
76. <a href="#">Maximum sum path negative</a>	<a href="#">Binary tree bottom to top level wise</a>
77. <a href="#">Find Kth smallest in BST</a>	<a href="#">Print Nodes of the same level</a>
78. <a href="#">BST with insertion order</a>	<a href="#">Linked list with inorder successor</a>
79. <a href="#">Binary tree sum of odd levels</a>	<a href="#">Linked list with preorder successor</a>
80. <a href="#">Level order without queue</a>	<a href="#">Linked list with postorder successor</a>
81. <a href="#">Print level with maximum nodes</a>	<a href="#">Binary tree to linked list</a>
82. <a href="#">Find all wrong pairs in a BST</a>	<a href="#">Is the binary tree BST</a>
83. <a href="#">Depth of tree from parent array</a>	<a href="#">Create tree from in and pre</a>
84. <a href="#">BST from doubly linked list</a>	<a href="#">Find root to node path</a>
85. <a href="#">Running Median</a>	<a href="#">Find distance between two nodes</a>
86. <a href="#">Maximum k integers using min heap</a>	<a href="#">SuperImpose Binary Tree</a>
87. <a href="#">Max heap and BST in one</a>	<a href="#">Is a sum possible along any path</a>
88. <a href="#">Merge N sorted arrays</a>	<a href="#">Remove duplicate infinite integer</a>
89. <a href="#">Find order of letters</a>	<a href="#">Find deepest level nodes</a>
90. <a href="#">Find longest interviewer chain</a>	<a href="#">Maximum sum path positive</a>
	<a href="#">Maximum sum path negative</a>
	<a href="#">Find Kth smallest in BST</a>
	<a href="#">BST with insertion order</a>
	<a href="#">Binary tree sum of odd levels</a>
	<a href="#">Level order without queue</a>
	<a href="#">Print level with maximum nodes</a>
	<a href="#">Find all wrong pairs in a BST</a>
	<a href="#">Depth of tree from parent array</a>

91. [Find local minima](#)
92. [Separate words in sentence](#)
93. [Increasing decreasing tuple](#)
94. [Two numbers sum up to k](#)
95. [Two numbers sum up to k unsorted](#)
96. [Any numbers sum up to k iterative](#)
97. [Any numbers sum up to k recursive](#)
98. [Container loading recursive](#)
99. [Largest palindrome dynamic](#)
100. [Longest common subsequence](#)
101. [Gold coins in pots game](#)
102. [Partition array equal sum](#)
103. [Find subset with given average](#)
104. [Share price max profit recursive](#)
105. [Increasing array subsequence](#)
106. [Distributed doubly linked list sum](#)
107. [Distributed node sum of](#)

[Depth of tree from parent array](#)

[BST from doubly linked list](#)

## Multithreading

[Distributed doubly linked list sum](#)

[Distributed node sum of tree](#)

[Distributed circular linked list sum](#)

## Graph

[Find order of letters](#)

[Find longest interviewer chain](#)

## Divide and conquer

[Find local minima](#)

## Dynamic programming

[Two numbers sum up to k](#)

[Two numbers sum up to k unsorted](#)

[Any numbers sum up to k iterative](#)

[Any numbers sum up to k recursive](#)

[Container loading recursive](#)

[Largest palindrome dynamic](#)

[Longest common subsequence](#)

[Gold coins in pots game](#)

[Partition array equal sum](#)

[Find subset with given average](#)

[Share price max profit recursive](#)

[Increasing array subsequence](#)

108. [tree](#)  
[Distributed circular](#)  
[linked list sum](#)



86 people like this. [Sign Up](#) to see what your friends like.

### 3 Comments



Add a comment...



**Raghu Talluri** ·

Works at Student

These are really good problems. I am learning to code by solving

[Like](#) · [Reply](#) · Jul 5, 2015 1:39pm



**Sentayehu Bekele** ·

Admas University College

Determine the big O-h notation, best case, average case and worst case for array copy, array split, array merge, and array shuffle algorithms

[Like](#) · [Reply](#) · May 7, 2015 4:44am



**Zaki Shaheen**

Dude, good website. But please get the damn ads off. I'd rather have no ads and malware.

[Like](#) · [Reply](#) · 2 · Feb 7, 2015 7:13am

