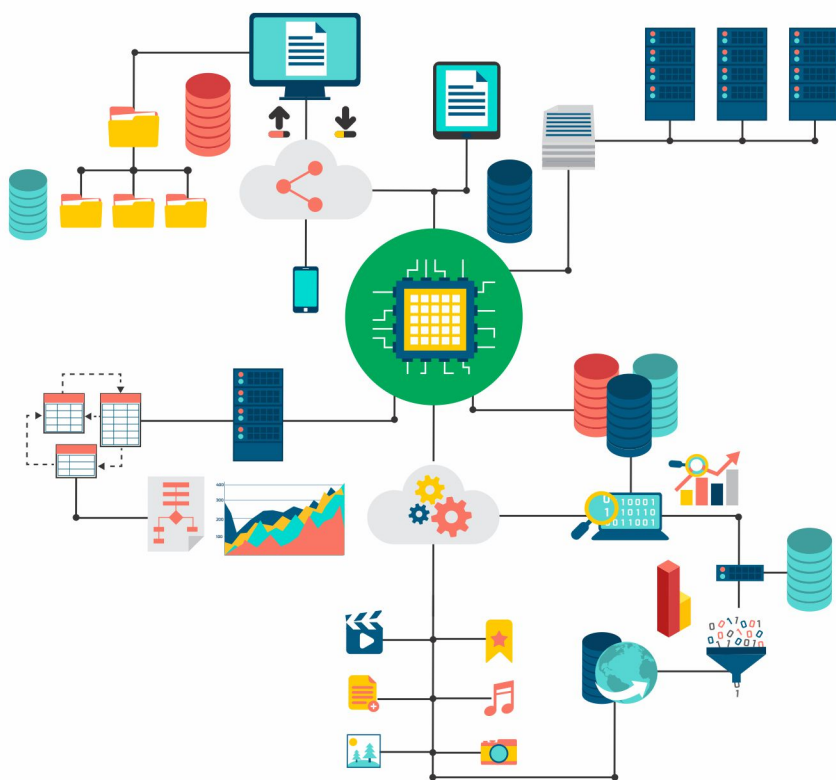


DATA STRUCTURES & ALGORITHMS

INTERVIEW QUESTIONS YOU'LL MOST LIKELY BE ASKED



277

Interview Questions



VIBRANT
PUBLISHERS

Job Interview Questions Series

DATA STRUCTURES & ALGORITHMS

**INTERVIEW QUESTIONS
YOU'LL MOST LIKELY BE ASKED**

277

Interview Questions



VIBRANT
PUBLISHERS

Data Structures And Algorithms

Interview Questions You'll Most Likely Be Asked

© 2021, By Vibrant Publishers, USA. All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior permission of the publisher.

ISBN-10: 1-946383-06-6

ISBN-13: 978-1-946383-06-8

Library of Congress Control Number: 2011921388

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. The author has made every effort in the preparation of this book to ensure the accuracy of the information. However, information in this book is sold without warranty either expressed or implied. The Author or the Publisher will not be liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Vibrant Publishers books are available at special quantity discount for sales promotions, or for use in corporate training programs. For more information please write to **bulkorders@vibrantpublishers.com**

Please email feedback / corrections (technical, grammatical or spelling) to **spellerrors@vibrantpublishers.com**

To access the complete catalogue of Vibrant Publishers, visit **www.vibrantpublishers.com**

Table of Contents

Data Structures

chapter 1	General Concepts Of Data Structures	5
chapter 2	Arrays	19
chapter 3	Stacks	25
chapter 4	Queues	31
chapter 5	Lists	35
chapter 6	Hash Data Structures	49
chapter 7	Trees	59
chapter 8	Sets	77
chapter 9	Graphs	83

Algorithms

chapter 10	General Concepts Of Algorithms	101
chapter 11	Sorting Algorithms	123
chapter 12	Search Algorithms	143
chapter 13	Huffman Coding	157
	HR Questions	165
	Index	196

This page is intentionally left blank

Data Structures And Algorithms

Review these typical interview questions and think about how you would answer them. Read the answers listed; you will find best possible answers along with strategies and suggestions.

This page is intentionally left blank.

Data Structures

This page is intentionally left blank.

Chapter 1

General Concepts Of Data Structures

1: What are Data Structures?

Answer:

Data Structures are a method of structured or organized data storage which is easier to manage. Data usually has 2 aspects – the definition and its implementation. Data structures make sure they keep both the aspects separate. The definition is made available for different implementations for different programs or usages. A data structure, as the name suggests, stores the information in a structured way so that it can be easily created, viewed and managed. This involves creating complex data types which are a combination of various basic data types. One example can be a customer data type which

will have a CustomerId of type Integer, Customer Name of type String and Address of type String.

2: What are the advantages of using Data Structures?

Answer:

Data structures help store the information in a structured way. This makes data storage and retrieval much easier than the conventional sequential storage. Data structures keep the data away from its implementation. This involves Data abstraction and encapsulation which are very important for code efficiency. Not only data, their relation can also be stored in data structures. One efficient way of implementing data structures are the databases. Data structures are also used for indexing data.

3: What is a data object?

Answer:

Data objects are the entities that actually contain the information. Data objects are the implementation of the data structures defined by the programmer. Data objects can access the methods and information stored in the data structure to store, process and retrieve information. It can be of type a complex structure or an array or an object in object oriented programming. A data object exists in the computer's memory till it is garbage collected whereas the data structure is accessed only when the object is being created. In simple terms, data structure is the data definition while data objects are its

implementations.

4: What are data types?

Answer:

Data types define what kind of information is stored in a variable or data member. There are primitive data types such as integer, character and Boolean and there are complex data types such as arrays data structures. Most of the programming languages require the variables to be declared before they are accessed. This helps in memory allocation depending on the data type of the variable. For example, an integer data type may require 4 bytes of memory while a float may require 8 bytes, depending on the programming language used. Complex data types are made of two or more primitive data types combined.

5: What are the different types of Data Structures?

Answer:

Data Structures can be categorized in many ways, though broadly, they can be categorized as Non-Linear and Linear data structures. Linear data structures are those that are sequential, like lists, arrays, stacks and queues. They are stored and can be accessed sequentially. Non-Linear data structures are objects or information that is not stored in an order. Graphs and trees are best example of non-linear data structures. While sequential data is easier to manage, non-linear data is not so easy. But many real-time solutions require non-linear data structures to

be implemented such as hierarchical data, geographical positioning and games.

6: Explain the Linear Data Structures.

Answer:

When information is stored in a sequential manner, it is easy to store and manage. Data structures, stored in an order or sequentially, are called linear data structures. Arrays, Lists, Queue, Stack, and even files are stored sequentially. While arrays and other data structures can be accessed directly with the position marker, files are accessed sequentially. Arrays can be single dimensional or multi-dimensional. But they are stored sequentially in the memory. For example, if there is an array of numbers $\text{num}[5] = \{2, 5, 7, 6, 1\}$ it is stored sequentially in the memory as

Num

(Position) 0	1	2	3	4
(Value) 2	5	7	6	1

This makes sure that linear data structures can be created and managed using pointers.

7: Explain the non-linear data structures.

Answer:

Non-linear information implies that the information does not follow a specific pattern for storage. But they can be related in other ways. Hierarchical information such as tree pattern or

geographical information that depends on positions can never be sequential. But they can be relative. Non-linear data structures are supported by most of the programming languages for implementing graphics, images, global positioning, location mapping and inheritance. These concepts are crucial to represent many real-time entities while developing applications and programs.

8: What are the basic operations possible in Data Structures?

Answer:

Every data structure allows some basic operations such as inserting, deleting, updating, traversing, searching, and sorting. Insert operations can be allowed in the beginning, end or the middle of the data structure based on the type of data structure. Similarly, deletion can also be allowed in the beginning, end or the middle of the data structure. Even though some data structures such as stacks and queues are very strict about inserting and deleting information, traversing and sorting the data structures work more or less similarly for all. Traversing and sorting are possible only because the data is stored sequentially.

9: What is a node?

Answer:

A node is the basic form of data structure. It basically consists of a member variable which holds the value and the address of the next node. The address part will be null for the last element

of the data structure. A node will allow all the basic operations of a data structure since the address to the next data node is stored in each node. With pointers, these dynamically allocated nodes can be easily accessed and traversed through for effecting the various operations such as insertion, deletion, updating, and sorting. An array is built by linking the nodes with the address element. There can be 2 addresses in a node where one address points to the previous node and the other address points to the next node.

10: What are Primitive Data types?

Answer:

Data types provide more information about the data. Most of the programs allow creating new data types which are implemented as enumerations and the values become the constants in the program. This makes the program more readable and understandable. For example, if you create a new data type for weekday with allowed values Sunday, Monday, Tuesday, Wednesday, Thursday, Friday and Saturday, these values are treated as constants instead of using switches or if-else constructs to check the value. Standard Primitive data types allowed in most of the languages are integer, character, Boolean, real and set. Integer data type allows storing number values ranging between negative and positive depending on the programming language. Real type allows storage of a subset of real numbers. Boolean allows true and false values. Char typically allows all alphabets, space, and a few special

characters depending on the language used.

11: Explain the record structure.

Answer:

A set of related information can be stored as a record. It can be considered as a complex data structure stored sequentially.

While the different elements of each entity are stored in columns, one entity comprises of a record. For example, the information regarding a student can be considered as the data structure and each student's detail can be considered as a record.

```
Type Student {
    StudName String;
    StudId int;
    DateOfBirth date;
    StudSex char;
    StudMaritalStatus char;
}
```

When you collect the student information, it will be stored in the following manner

Structure - Record No	Stud Name	Stud Id	DateOfBirth DD/MM/YYYY	StudSex M / F	StudMaritalStatus S / M / W / D
1	Smith Allen	1	10/9/1982	M	S
2	Mary Lynda	2	09/02/1982	F	M
3	Alex George	3	20/01/1983	M	S

The information regarding each student is stored as a record. Records make data storage and retrieval much easier to manage.

12: What is a file? How is it different from a record?**Answer:**

Files store data sequentially in the hard disk. Every information that we want to store permanently or make persistent is stored as a named file in the hard disk or in a database as records. The main difference between files and database are that files store information sequentially while the records are stored as structured information. A file will have the same data type information stored. But the record will have multiple data types as defined in the data structure. The length of a file can be dynamically allocated, based on the storage memory available on the disk or allotted to a particular user. For records, each record will have the maximum size allocated as per its definition that comprises of different data types.

13: Explain the difference between sequential and structured data.**Answer:**

Sequential data is easier to create but difficult to manage. Structured data is complex to create and manage but is the best when it comes to retrieval and processing. Sequential data can be considered as text files with no structure and structured data can be considered as data in tabular form or as records as

we get from the database. While text information is easier to create, retrieving specific information is very difficult with sequential data. The entire file has to be loaded and specific search has to be performed to retrieve which will scan through the entire file. With structured data, it can be considered as information in tabular form. Specific columns or rows can be spotted and easily retrieved which makes data retrieval more efficient. But creating and managing structured data requires experienced professionals.

14: Explain the different sorting methods commonly used to sort arrays.

Answer:

Sorting the information available is an important aspect of data processing. Sorting is nothing but rearranging the information available in a particular order. When it comes to arrays, sorting the information can be done in many ways. The commonly used techniques to sort an array are insertion sort, bubble sort, selection sort, quick sort, tree sort, merge sort, and shell sort. Each sorting method has its own logic and algorithm and involves rearranging the array elements in the ascending or descending order. While some sorting techniques involve rearranging every element of the array during the process, some techniques use methods to find the lowest first and position it and consider the rest of the array during each pass.

15: Explain Selection sort.**Answer:**

In Selection sort, the basic idea is to find the smallest number and position it in the first array index and then move ahead for the next one and position it in the second and so on till all the elements are refilled. This is if you want to sort in ascending order. If you want to order in descending order, start with the biggest number first. Find the following example, if you have an array with 5 elements and want to sort it in ascending order:

Original Array -> 5, 3, 1, 9, 2

1st pass -> 1, 3, 5, 9, 2

2nd Pass -> 1, 2, 5, 9, 3

3rd Pass -> 1, 2, 5, 9, 3

4th Pass -> 1, 2, 3, 9, 5

5th Pass -> 1, 2, 3, 5, 9 – which is the final sorted array.

16: Explain Bubble sort.**Answer:**

Bubble sort works with swapping the adjacent array elements. It takes longer time as the array gets longer since for each pass, all the adjacent elements have to be checked and swapped if necessary. For an array with 5 elements and want to bubble sort it in ascending order:

Original Array -> 5, 3, 1, 9, 2

1st pass ->

5, 3, 1, 9, 2 -> 3, 5, 1, 9, 2 -> 3, 1, 5, 9, 2 -> 3, 1, 5, 2, 9

2nd pass ->

3, 1, 5, 2, 9 -> 1, 3, 5, 2, 9 -> 1, 3, 5, 2, 9 -> 1, 3, 2, 5, 9 -> 1, 3, 2, 5, 9

3rd Pass ->

1, 3, 2, 5, 9 -> 1, 3, 2, 5, 9 -> 1, 2, 3, 5, 9 -> 1, 2, 3, 5, 9 -> 1, 2, 3, 5, 9

If selection sort required 5 passes, bubble sort got completed in 3rd pass. In each pass, the largest number became the last element and the next one, the next largest and so on.

17: What are linked lists?

Answer:

A linked list is a collection of connected nodes. Each node will have a data element and the address that links to the next node. The first node will have no data. It will only have the link to the next node. The last node will have only data and the link to the next node will be null. The first node is called the head and the last node is the tail. Linked lists help the programmer to dynamically create arrays. There are 3 types of linked lists – singly-linked, doubly-linked and circular-linked list. The singly-linked list will have only one link in each node that points to the next node. The doubly-linked list will have the links or the pointers to the previous node and next node. The circular-linked list will have the last node pointing to the first node's address of the linked list. It can be singly or doubly linked.

18: Explain Binary Search Tree.

Answer:

Binary Search Trees will have a root with 2 nodes attached. The nodes can be sub-trees or simply nodes. The left side nodes will have values less than the root and the right side nodes will have values greater than the root. There can be balanced and unbalanced binary trees. The balanced trees are when each node has either no sub-node or has 2 sub-nodes. Unbalanced binary trees are when a node has only one sub-node either to its left or right.

19: Explain the different algorithm patterns used with data structures.

Answer:

Data Structures form the key element of programming for all programmers. One needs to understand some basic algorithms commonly used with data structures which will help handling all types of complex data objects. Since data structures can be implemented in different ways in different programs, understanding the basic algorithms will help implement them better in real-time situations. Graph algorithms help a programmer handle Graph related data easily. Dynamic programming algorithms help in dynamic allocation of memory for arrays and other data objects. Searching and sorting algorithms are required everywhere. Applying the various number theories and various mathematical formulas such as prime factorization, factorials etc. will help creating applications involving complex mathematical equations.

Geometrical and network algorithms help in establishing complex networks easily.

20: What are the practical applications of data structures?

Answer:

Data structures are the basic idea behind many complex software solutions that. The concept of data structures are used in building operating systems, numerical analysis, database management system, graphics, artificial intelligence, statistical analysis applications, compiler design and simulations. For example, memory allocations use the heap and stack concepts to store and retrieve information in the computer memory. Any business application that requires complex matrices and graphs will require the implementation of data structures. The database is the basic requirement for most of the business applications as it stores a large amount of structured data which is easier to process and retrieve.

This page is intentionally left blank.

Job Interview Questions Series

DATA STRUCTURES & ALGORITHMS

INTERVIEW QUESTIONS
YOU'LL MOST LIKELY BE ASKED



Data Structures & Algorithms Interview Questions You'll Most Likely Be Asked is a perfect companion to stand ahead above the rest in today's competitive job market. Rather than going through comprehensive, textbook-sized reference guides, this book includes only the information required immediately for job search to build an IT career. This book puts the interviewee in the driver's seat and helps them steer their way to impress the interviewer.

The following is included in this book:

- 200 Data Structures & Algorithms Interview Questions, Answers and proven strategies for getting hired as an IT professional
- Dozens of examples to respond to interview questions
- 77 HR Questions with Answers and proven strategies to give specific, impressive, answers that help nail the interviews
- 2 Aptitude Tests download available on www.vibrantpublishers.com



www.vibrantpublishers.com

