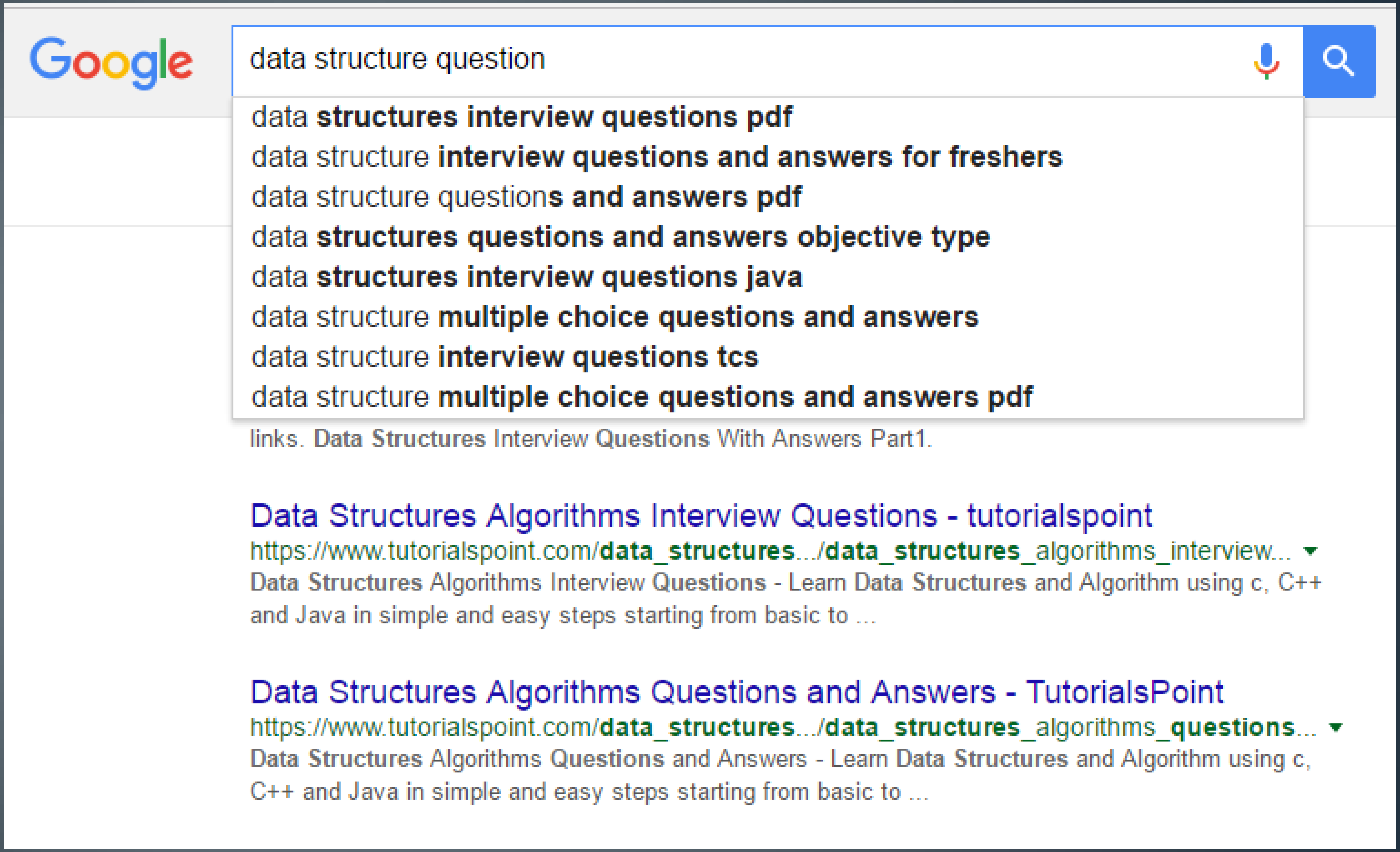
Introduction to Data Structures

2nd Semester

Batch-2019

BS(Computer Science)



data structure question

data **structures interview questions pdf**

data structure **interview questions and answers for freshers**

data structure questions **and answers pdf**

data **structures questions and answers objective type**

data **structures interview questions java**

data structure **multiple choice questions and answers**

data structure **interview questions tcs**

data structure **multiple choice questions and answers pdf**

links. **Data Structures** Interview **Questions** With Answers Part1.

Data Structures Algorithms Interview Questions - tutorialspoint

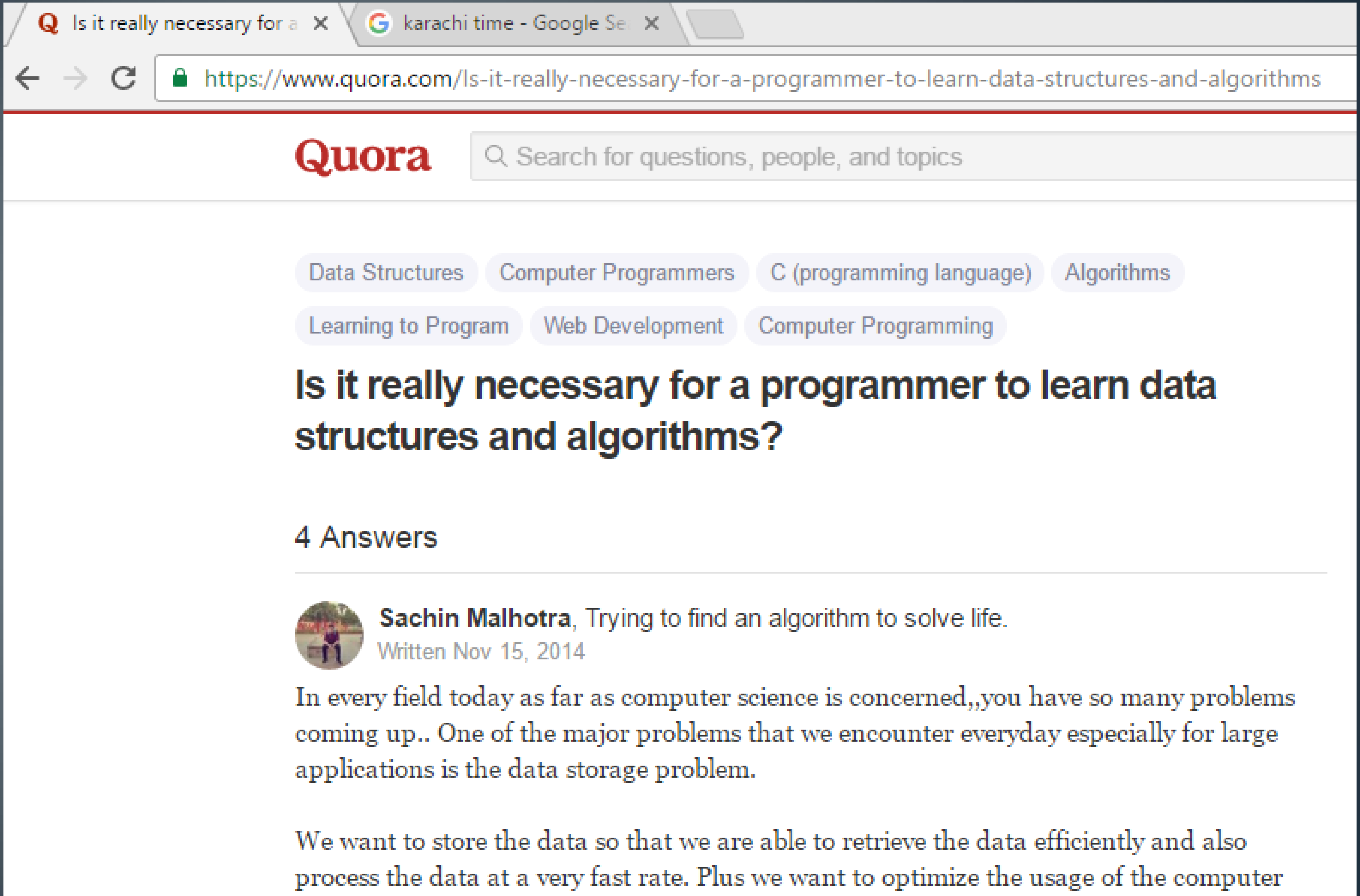
https://www.tutorialspoint.com/**data_structures**.../**data_structures**_algorithms_interview...

Data Structures Algorithms Interview **Questions** - Learn **Data Structures** and Algorithm using c, C++ and Java in simple and easy steps starting from basic to ...

Data Structures Algorithms Questions and Answers - TutorialsPoint

https://www.tutorialspoint.com/**data_structures**.../**data_structures**_algorithms_**questions**...

Data Structures Algorithms **Questions** and Answers - Learn **Data Structures** and Algorithm using c, C++ and Java in simple and easy steps starting from basic to ...



Is it really necessary for a
karachi time - Google Se
https://www.quora.com/Is-it-really-necessary-for-a-programmer-to-learn-data-structures-and-algorithms
Quora
Search for questions, people, and topics
Data Structures
Computer Programmers
C (programming language)
Algorithms
Learning to Program
Web Development
Computer Programming

Is it really necessary for a programmer to learn data structures and algorithms?

4 Answers

Sachin Malhotra, Trying to find an algorithm to solve life.
Written Nov 15, 2014

In every field today as far as computer science is concerned,,you have so many problems coming up.. One of the major problems that we encounter everyday especially for large applications is the data storage problem.

We want to store the data so that we are able to retrieve the data efficiently and also process the data at a very fast rate. Plus we want to optimize the usage of the computer

Goals

"I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important. **Bad programmers worry about the code. Good programmers worry about data structures and their relationships.**"

Linus Torvalds, 2006

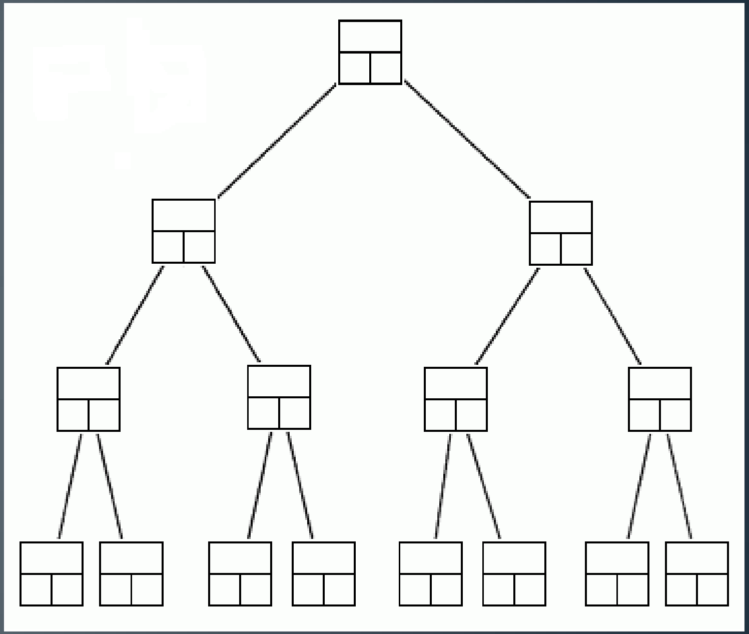
data structure

- A **data structure** is a specialized format for organizing and storing **data**. General **data structure** types include the array, the file, the record, the table, the tree, and so on. Any **data structure** is designed to organize **data** to suit a specific purpose so that it can be accessed and worked with in appropriate ways.

Data Structures

A data structure is a scheme for organizing data in the memory of a computer.

Some of the more commonly used data structures include lists, arrays, stacks, queues, heaps, trees, and graphs.

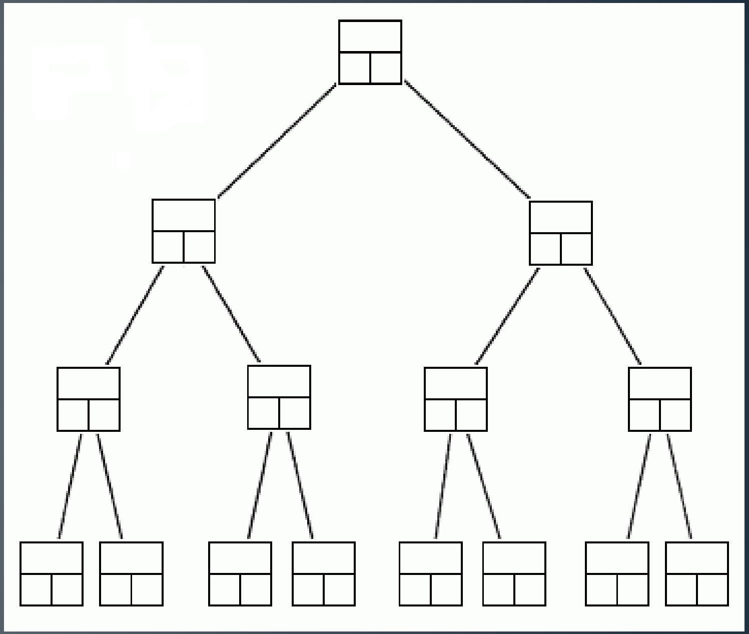


Binary Tree

Data Structures

The way in which the data is organized affects the performance of a program for different tasks.

Computer programmers decide which data structures to use based on the nature of the data and the processes that need to be performed on that data.



Binary Tree

Example: A Queue

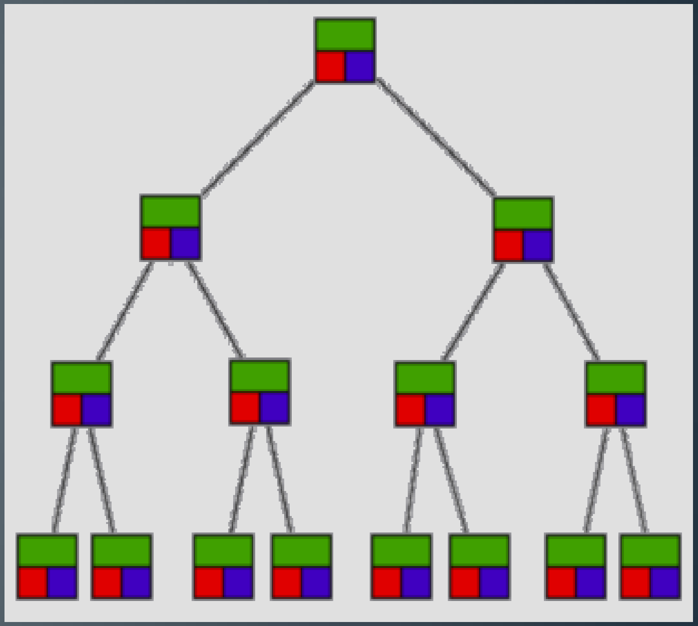
A *queue* is an example of commonly used simple data structure. A queue has beginning and end, called the *front* and *back* of the queue.

Data enters the queue at one end and leaves at the other. Because of this, data exits the queue in the same order in which it enters the queue, like people in a checkout line at a supermarket.

Example: A Binary Tree

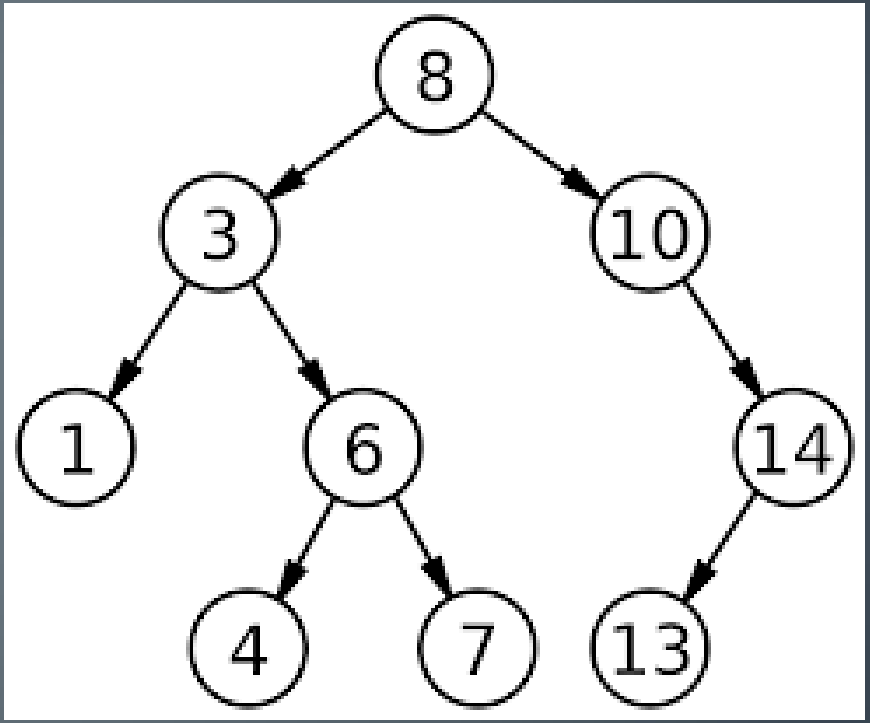
A *binary tree* is another commonly used data structure. It is organized like an upside down tree.

Each spot on the tree, called a *node*, holds an item of data along with a left pointer and a right pointer.



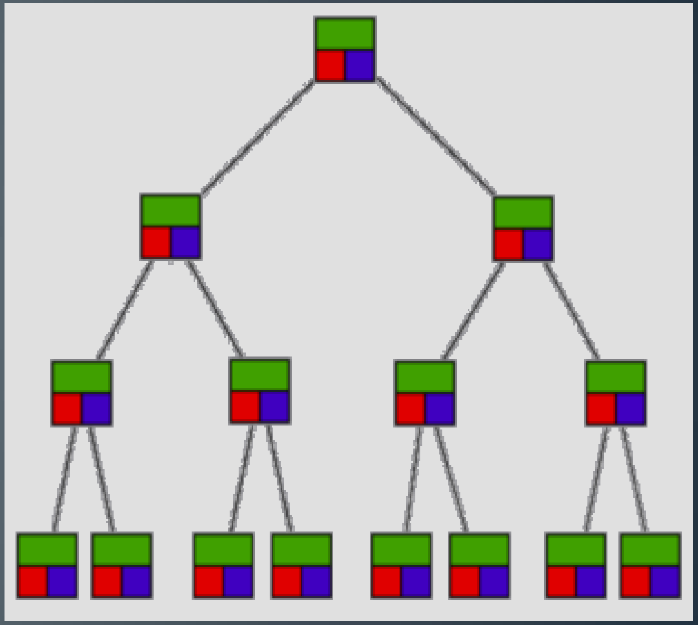
Binary Tree

Example: A Binary Tree



Example: A Binary Tree

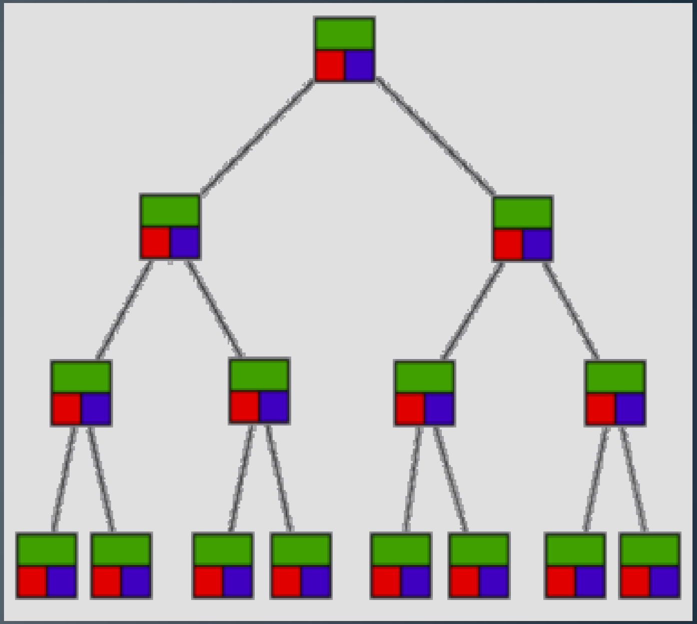
The pointers are lined up so that the structure forms the upside down tree, with a single node at the top, called the root node, and branches increasing on the left and right as you go down the tree.



Binary Tree

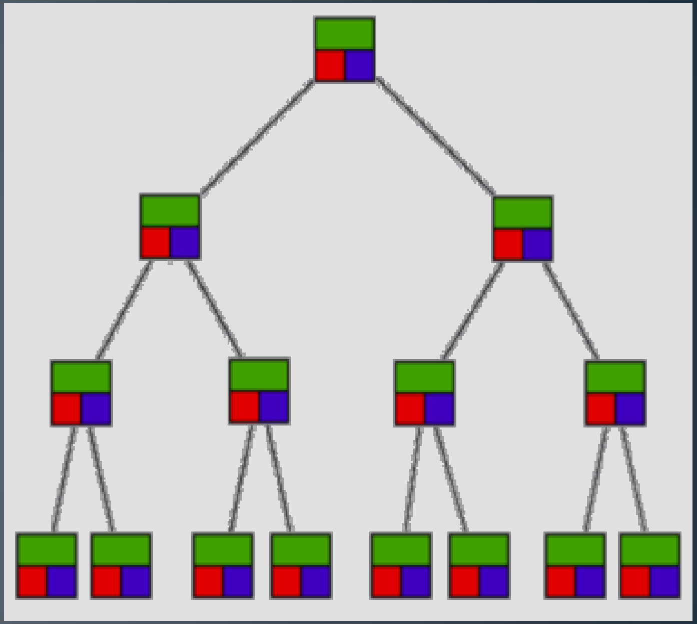
Choosing Data Structures

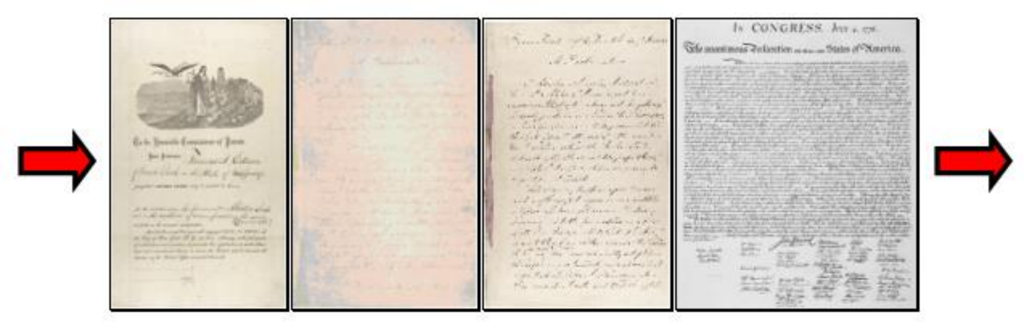
By comparing the queue with the binary tree, you can see how the structure of the data affects what can be done efficiently with the data.



Choosing Data Structures

A queue is a good data structure to use for storing things that need to be kept in order, such as a set of documents waiting to be printed on a network printer.



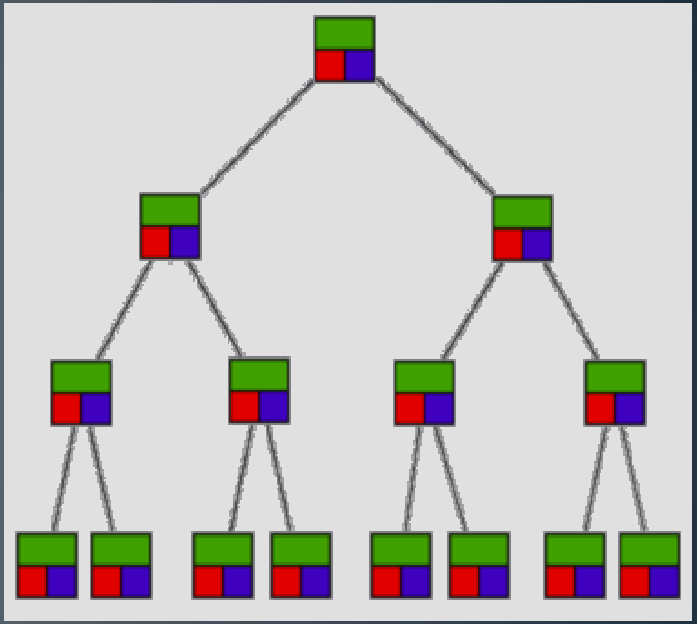


Choosing Data Structures

The jobs will be printed in the order in which they are received.

Most network print servers maintain such a *print queue*.

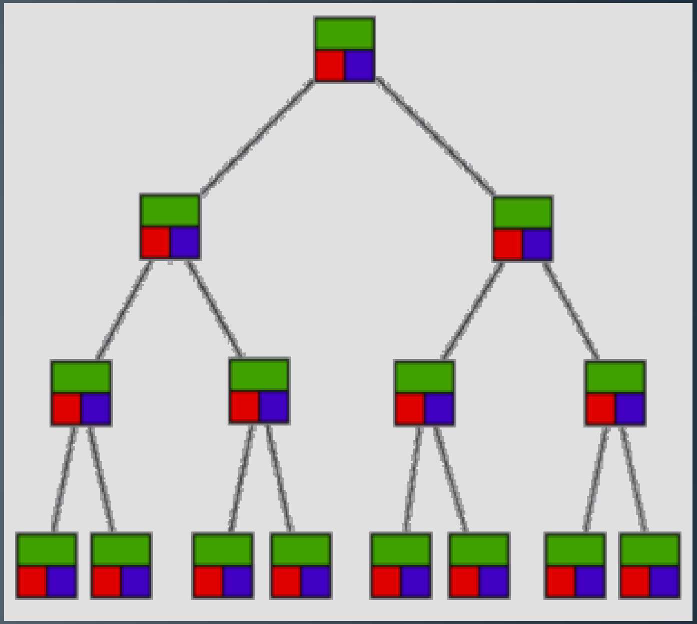
.



Choosing Data Structures

A binary tree is a good data structure to use for searching sorted data.

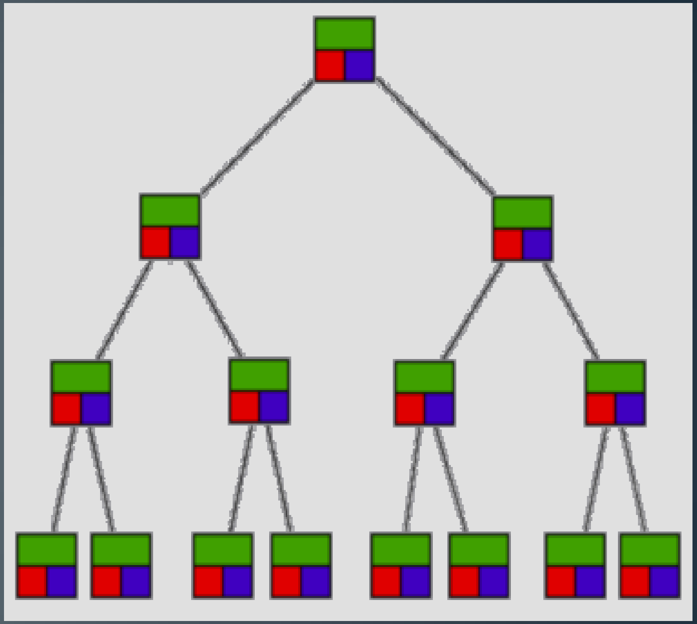
The middle item from the list is stored in the root node, with lesser items to the left and greater items to the right.



Choosing Data Structures

A search begins at the root. The computer either find the data, or moves left or right, depending on the value for which you are searching.

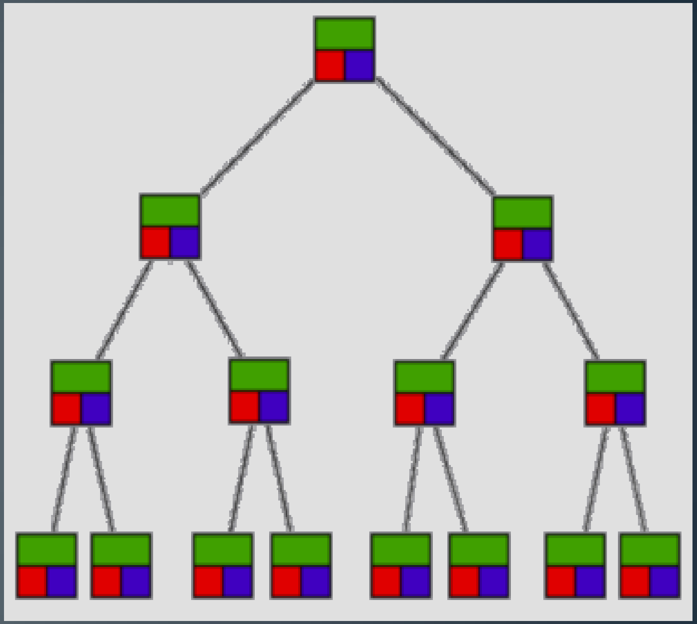
Each move down the tree cuts the remaining data in half.



Choosing Data Structures

Items can be located very quickly in a tree.

Telephone directory assistance information is stored in a tree, so that a name and phone number can be found quickly.



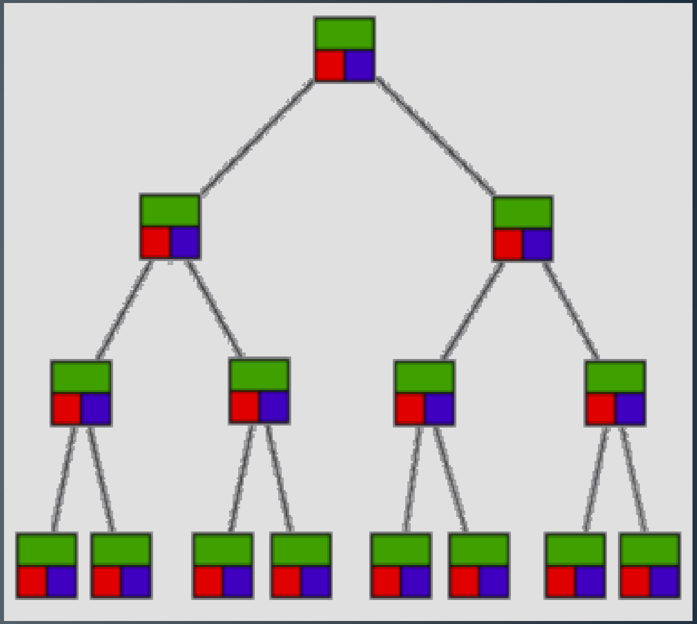
In CONGRESS

Choosing Data Structures

For some applications, a queue is the best data structure to use.

For others, a binary tree is better.

Programmers choose from among many data structures based on how the data will be used by the program.



Lists

A list is an ordered set of data. It is often used to store objects that are to be processed sequentially.

A list can be used
 to create a queue.

Arrays

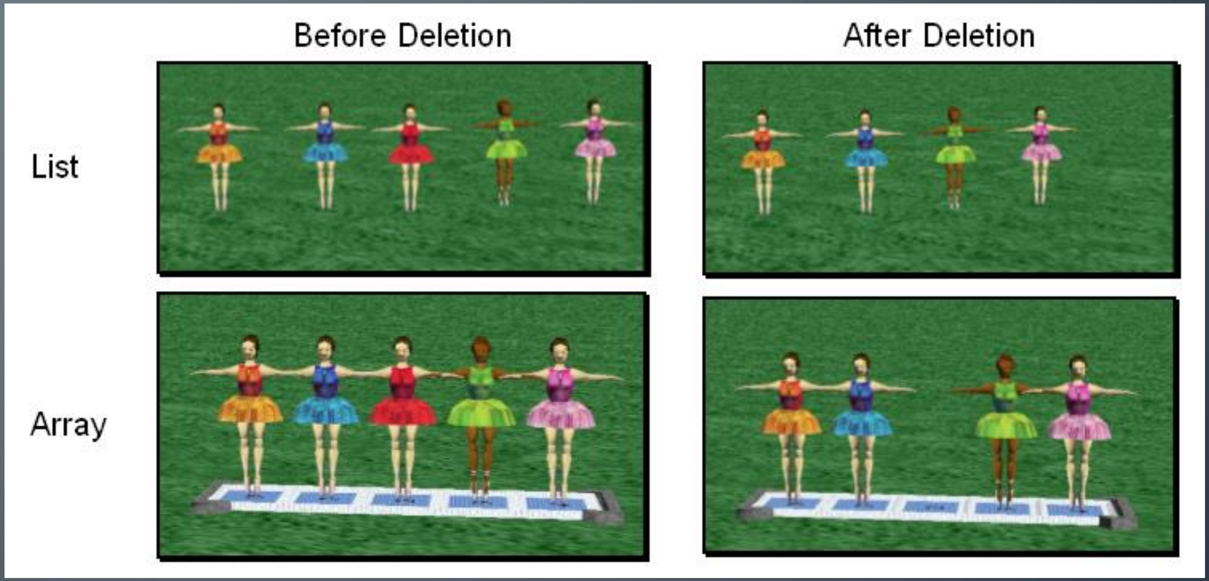
An array is an indexed set of variables, such as toy$_{[1]}$, toy$_{[2]}$, toy$_{[3]}$,… It is like a set of boxes that hold things.

A list is a set of items.

An array is a set of variables that each store an item.

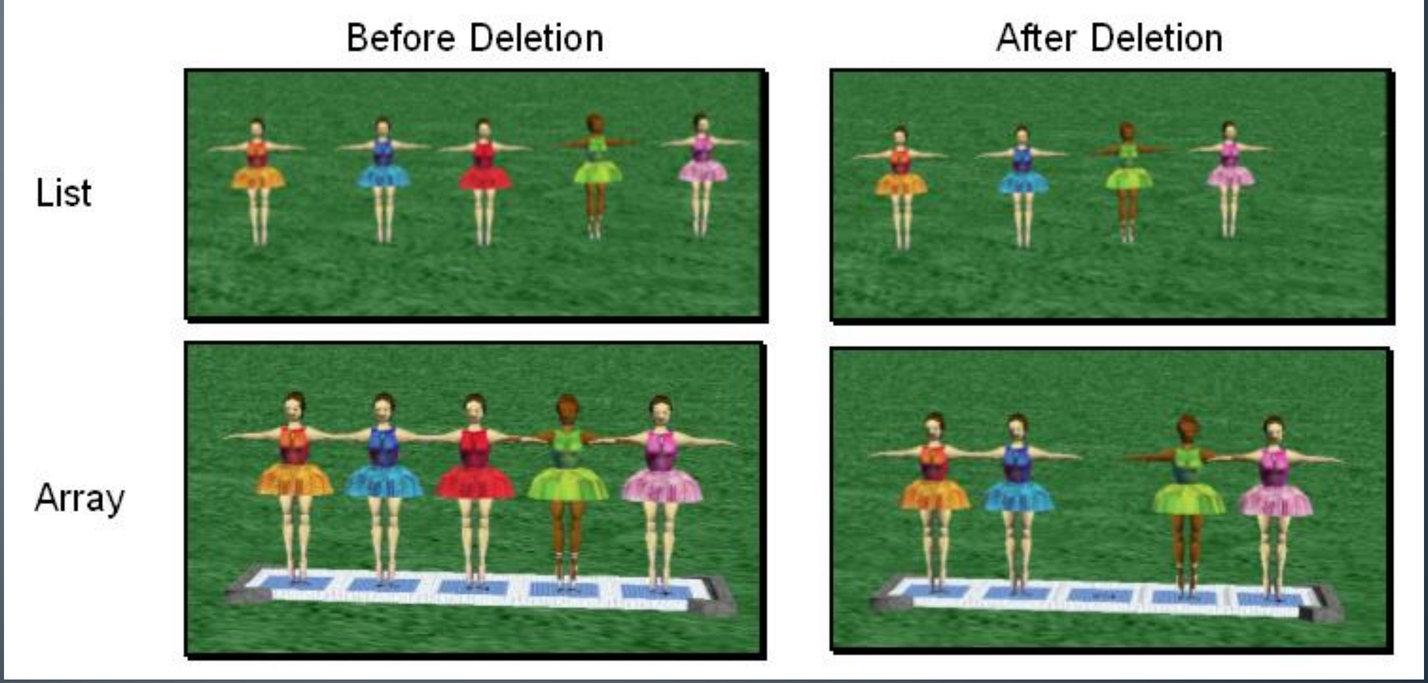
Arrays and Lists

You can see the difference between arrays and lists when you delete items.



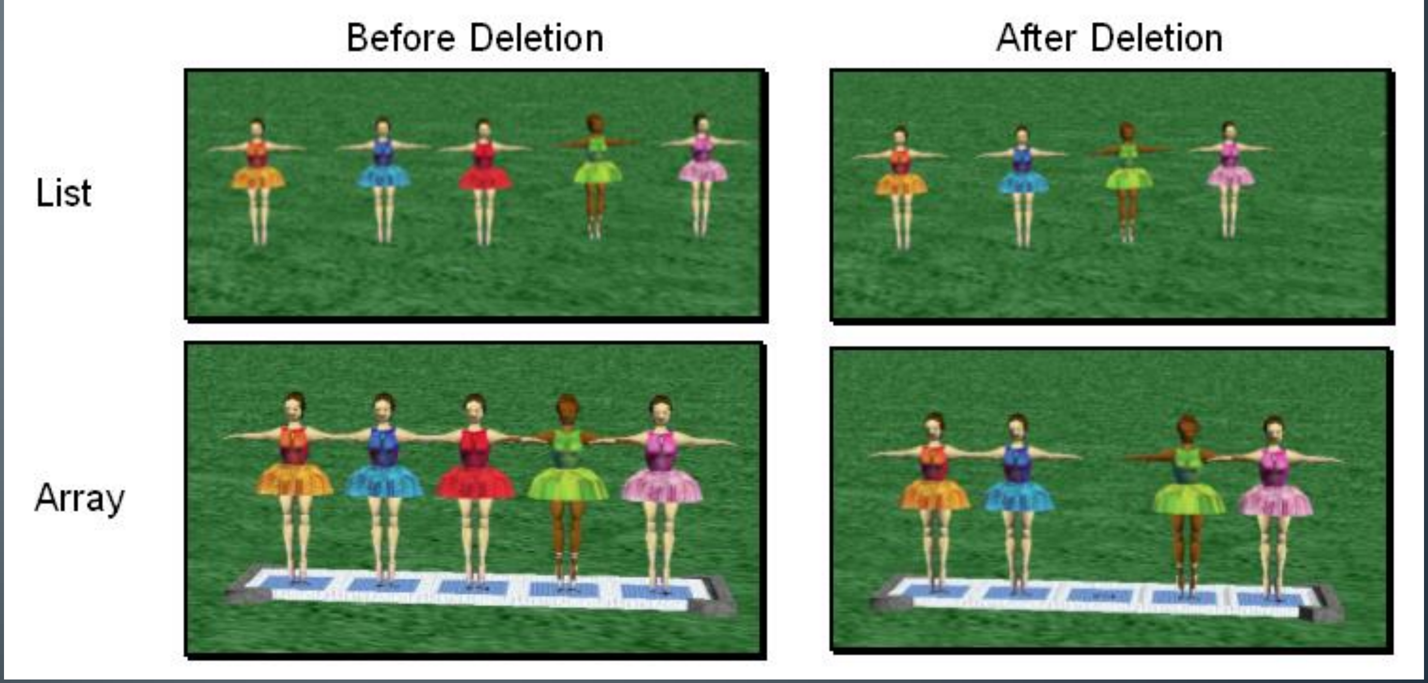
Arrays and Lists

In a list, the missing spot is filled in when something is deleted.



Arrays and Lists

In an array, an empty variable is left behind when something is deleted.


Before Deletion
After Deletion
List
Array