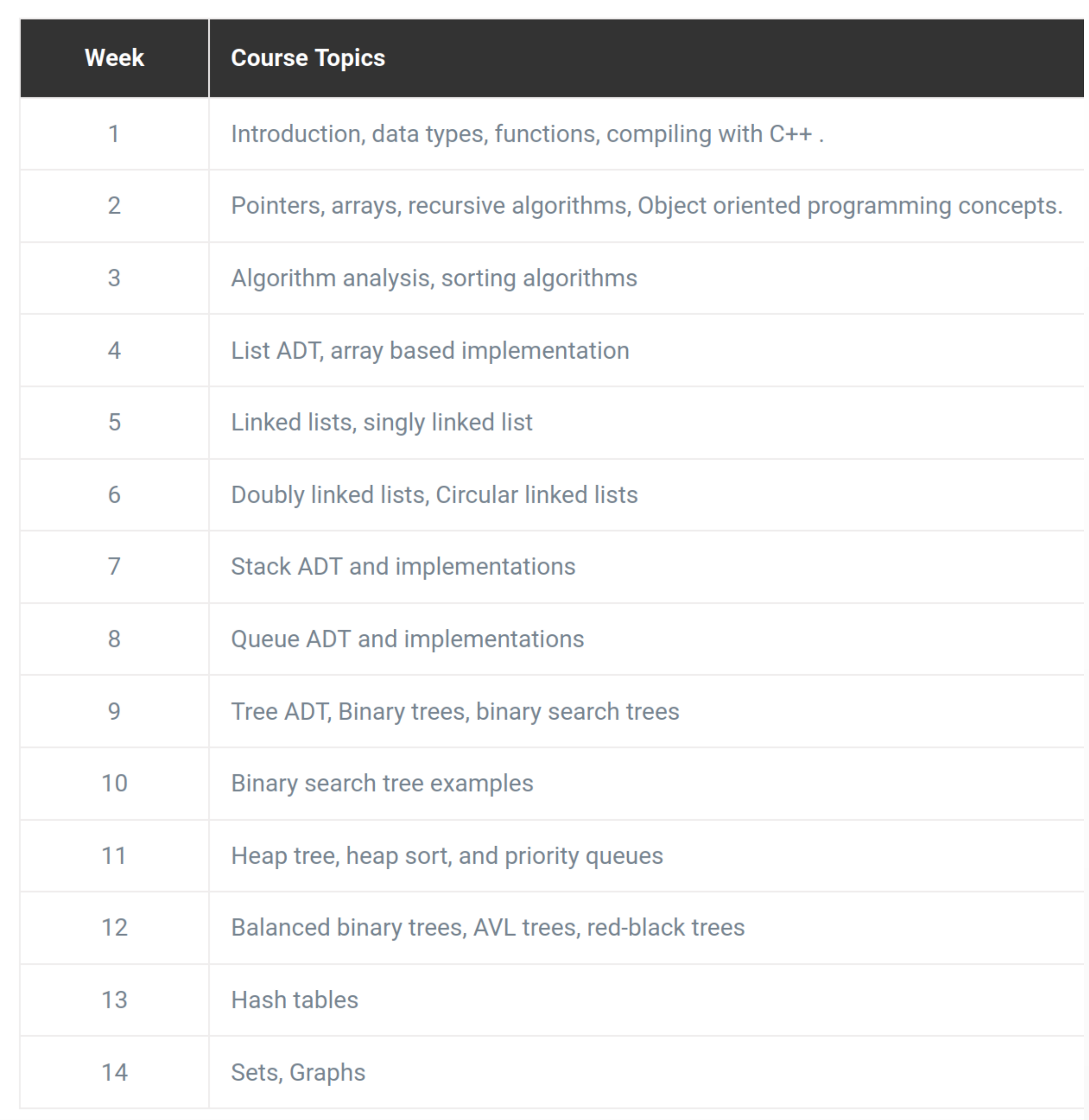
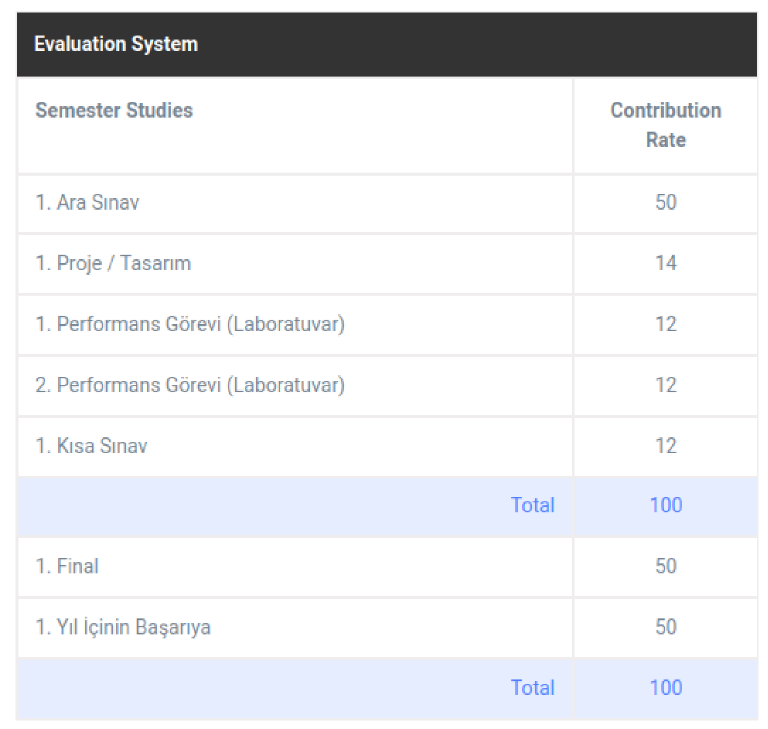
Data Structures and Algorithms

Introduction

Tentative course outline



Week	Course Topics
1	Introduction, data types, functions, compiling with C++ .
2	Pointers, arrays, recursive algorithms, Object oriented programming concepts.
3	Algorithm analysis, sorting algorithms
4	List ADT, array based implementation
5	Linked lists, singly linked list
6	Doubly linked lists, Circular linked lists
7	Stack ADT and implementations
8	Queue ADT and implementations
9	Tree ADT, Binary trees, binary search trees
10	Binary search tree examples
11	Heap tree, heap sort, and priority queues
12	Balanced binary trees, AVL trees, red-black trees
13	Hash tables
14	Sets, Graphs



Evaluation System

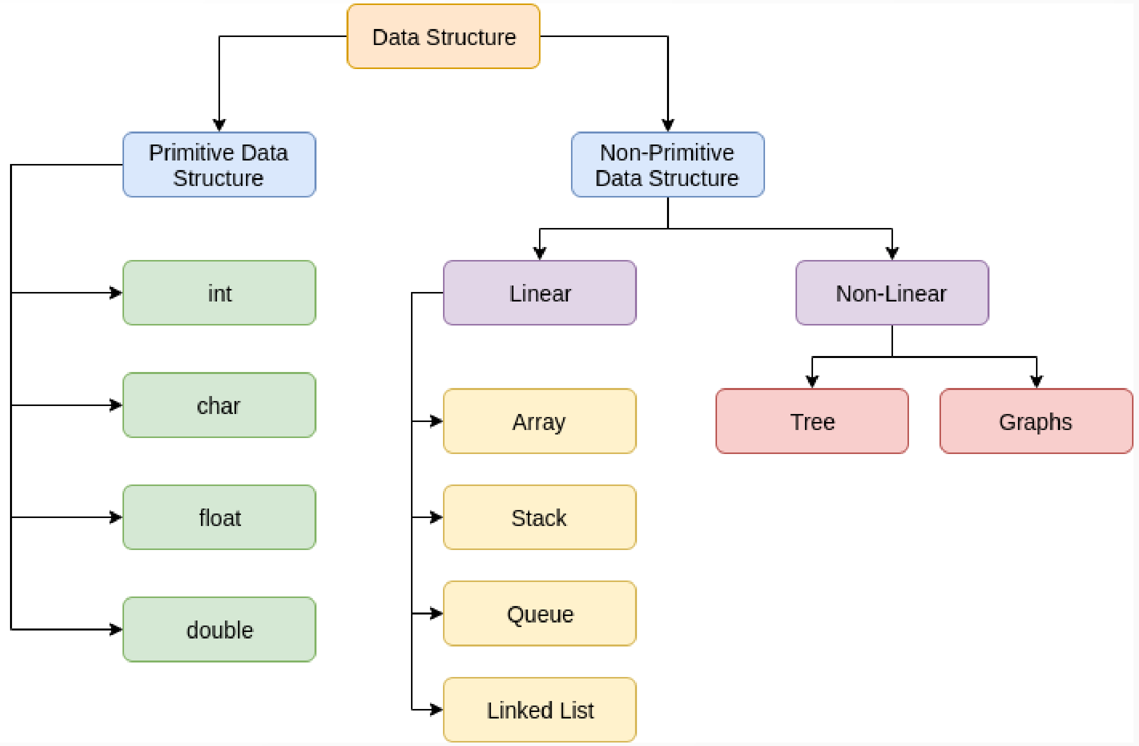
Semester Studies	Contribution Rate
1. Ara Sınav	50
1. Proje / Tasarım	14
1. Performans Görevi (Laboratuvar)	12
2. Performans Görevi (Laboratuvar)	12
1. Kısa Sınav	12
Total	100
1. Final	50
1. Yıl İçinin Başarıya	50
Total	100

Overview: Data structures

- In computer science, a data structure is a data organization, management, and storage format that enables efficient access and modification.
- More precisely, a data structure is a collection of data values, the relationships among them, and the functions or operations that can be applied to the data, i.e., it is an algebraic structure about data.
- Data structures serve as the basis for abstract data types (ADT).
- The ADT defines the logical form of the data type.
- The data structure implements the physical form of the data type

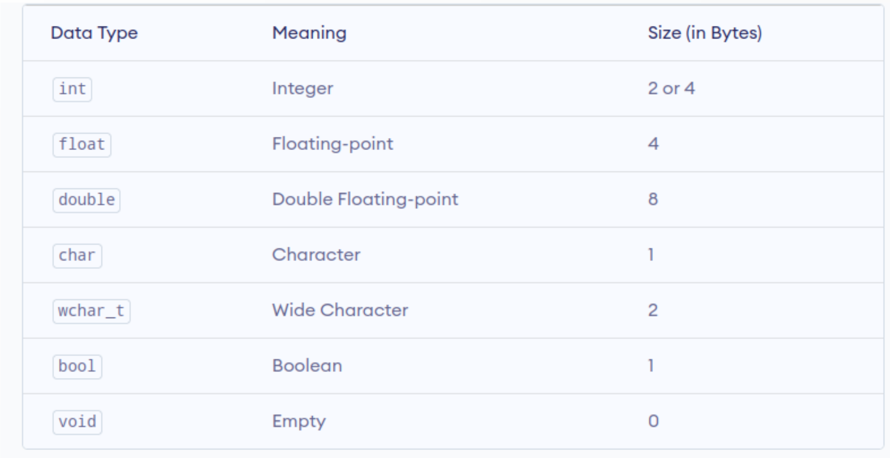
Overview: Data structures

- Data structures are primarily categorized into two parts:
- **Primitive Data Structures**
- These are the predefined way of storing data in the system. All sets of operations are pre-defined. Char, int, float, double are examples of primitive data structures.
- The data structures which are designed using primitive data structures are called non-primitive data structures. They are used to store a collection of data. It can be categorized into two parts:
- Linear data structure
- Non-Linear data structure



Overview: C++ programming and OOP

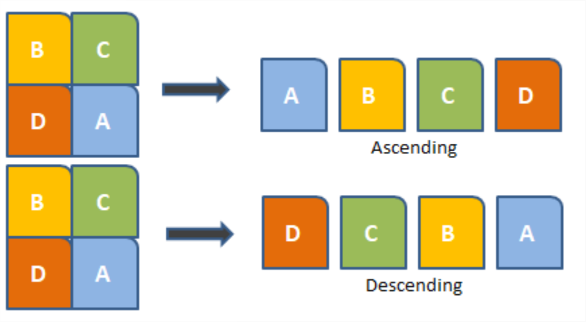
- C++ programming language will be used to implement and test the algorithm and datastructures
- C++ language is a low level language when compared to the languages such as C# and Java
- A lower level of language than C++ is the C language
- C doesn’t support OOP
- OOP is a programming methodology constructed around objects. OOP is associated with concepts such as class, object, Inheritance, Encapsulation, Abstraction, Polymorphism etc.



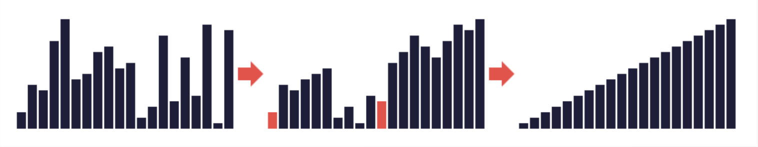
Data Type	Meaning	Size (in Bytes)
`int`	Integer	2 or 4
`float`	Floating-point	4
`double`	Double Floating-point	8
`char`	Character	1
`wchar_t`	Wide Character	2
`bool`	Boolean	1
`void`	Empty	0

Overview:Sorting algorithms

- Sorting Algorithms are methods of reorganizing a large number of items into some specific order such as highest to lowest, or vice-versa, or even in some alphabetical order.
- These algorithms take an input list, processes it (i.e, performs some operations on it) and produce the sorted list.
- The most common example we experience every day is sorting clothes or other items on an e-commerce website either by lowest-price to highest, or list by popularity, or some other order.



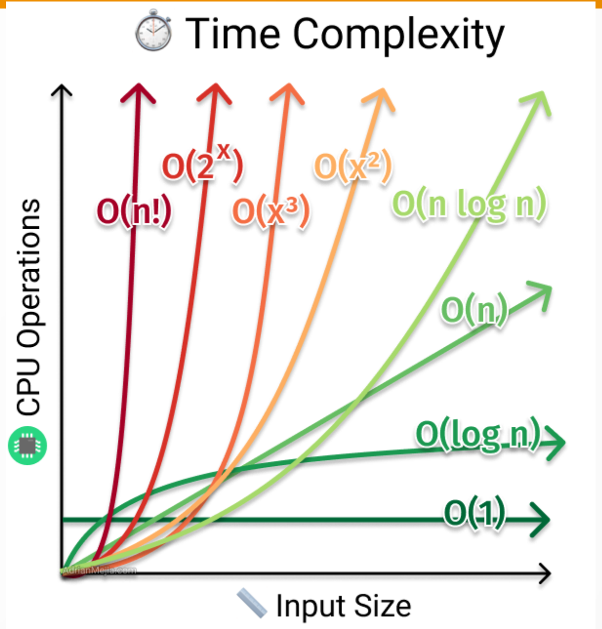
https://www.equestionanswers.com/c/c-sorting.php



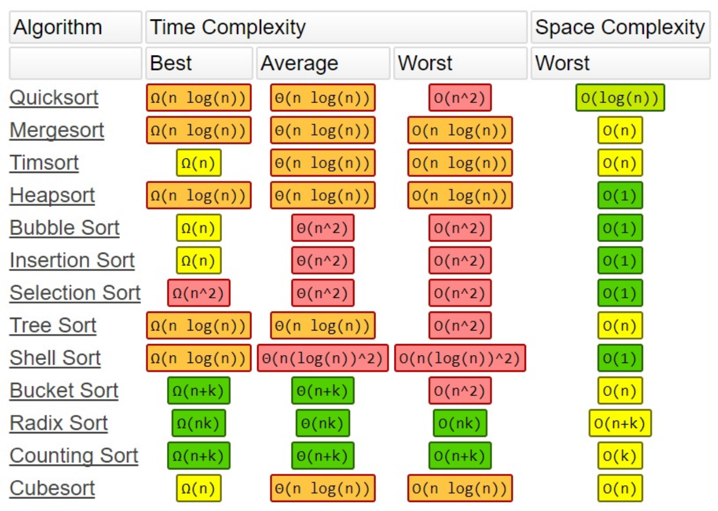
- Quick Sort
- Bubble Sort
- Merge Sort
- Insertion Sort
- Selection Sort
- Heap Sort
- Radix Sort
- Bucket Sort

Overview: Algorithm analysis

- Algorithmic complexity is a measure of how long an algorithm would take to complete given an input of size n.
- If an algorithm has to scale, it should compute the result within a finite and practical time bound even for large values of n.
- For this reason, complexity is calculated asymptotically as n approaches infinity.
- While complexity is usually in terms of time, sometimes complexity is also analyzed in terms of space, which translates to the algorithm's memory requirements
- Analysis of an algorithm's complexity is helpful when comparing algorithms or seeking improvements.
- It's important to note that we're concerned about the order of an algorithm's complexity, not the actual execution time in terms of milliseconds.
- Algorithmic complexity is also called complexity or running time
- There are various sorting algorithms such as: Merge sort, quick sort, selection sort etc. Although the objective of them is to sort an array the way their algorithms work differ in the way they use the CPU and Memory.



https://adrianmejia.com/how-to-find-time-complexity-of-an-algorithm-code-big-o-notation/



Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
Quicksort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(log(n))
Mergesort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(n)
Timsort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)
Heapsort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(1)
Bubble Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Insertion Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Selection Sort	Ω(n^2)	Θ(n^2)	O(n^2)	O(1)
Tree Sort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(n)
Shell Sort	Ω(n log(n))	Θ(n(log(n))^2)	O(n(log(n))^2)	O(1)
Bucket Sort	Ω(n+k)	Θ(n+k)	O(n^2)	O(n)
Radix Sort	Ω(nk)	Θ(nk)	O(nk)	O(n+k)
Counting Sort	Ω(n+k)	Θ(n+k)	O(n+k)	O(k)
Cubesort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)

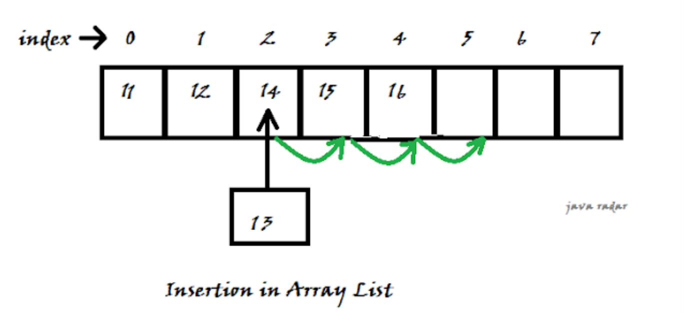
https://devopedia.org/algorithmic-complexity

Overview: Lists

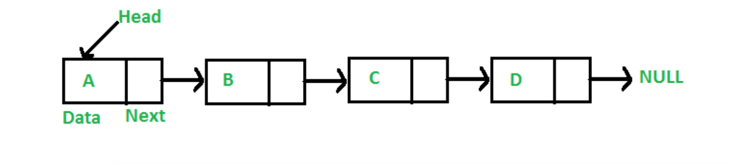
- In computer science, a list or sequence is an abstract data type that represents a finite number of ordered values, where the same value may occur more than once.
- An instance of a list is a computer representation of the mathematical concept of a tuple or finite sequence; the (potentially) infinite analog of a list is a stream.
- Lists are a basic example of containers, as they contain other values.
- If the same value occurs multiple times, each occurrence is considered a distinct item.
- Many programming languages provide support for list data types, and have special syntax and semantics for lists and list operations
- Array based implementation
- Linked (Node based) implementation

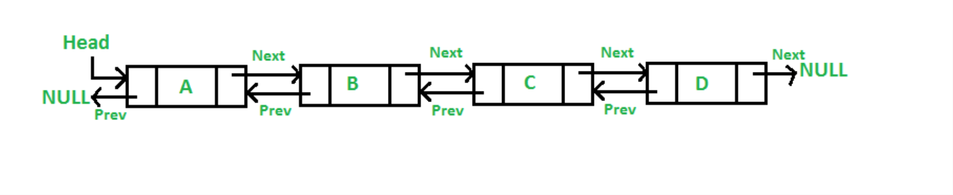
https://en.wikipedia.org/wiki/List_(abstract_data_type)

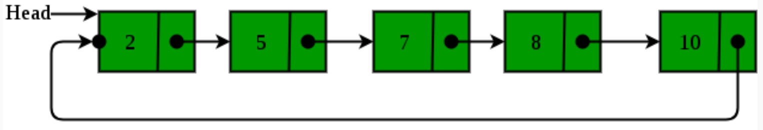
Array based list



Linked (Node based) list





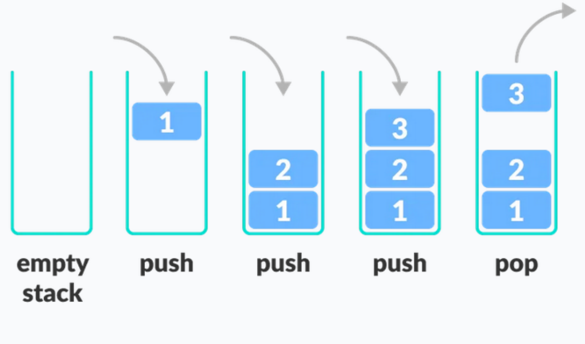


https://www.geeksforgeeks.org

Overview: Stacks

- A stack is a linear data structure that follows the principle of Last In First Out (LIFO).
- This means the last element inserted inside the stack is removed first.
- You can think of the stack data structure as the pile of plates on top of another.

Stack representation similar to a pile of plate

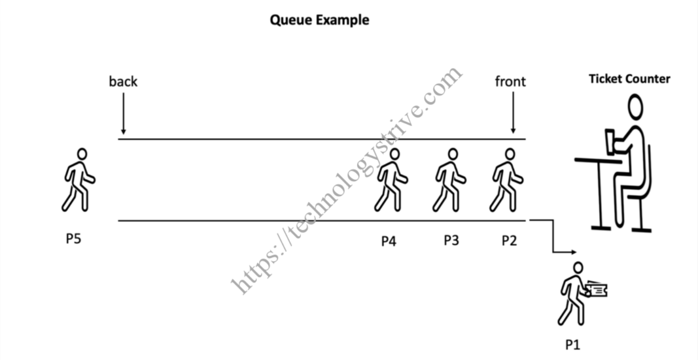


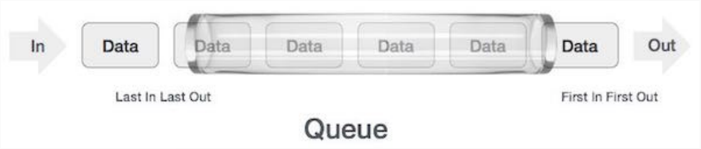
Stack Push and Pop Operations

https://www.programiz.com/dsa/stack

Overview: Queues

- Queue is an abstract data structure, somewhat similar to Stacks.
- Unlike stacks, a queue is open at both its ends.
- One end is always used to insert data (enqueue) and the other is used to remove data (dequeue).
- Queue follows First-In-First-Out methodology, i.e., the data item stored first will be accessed first.

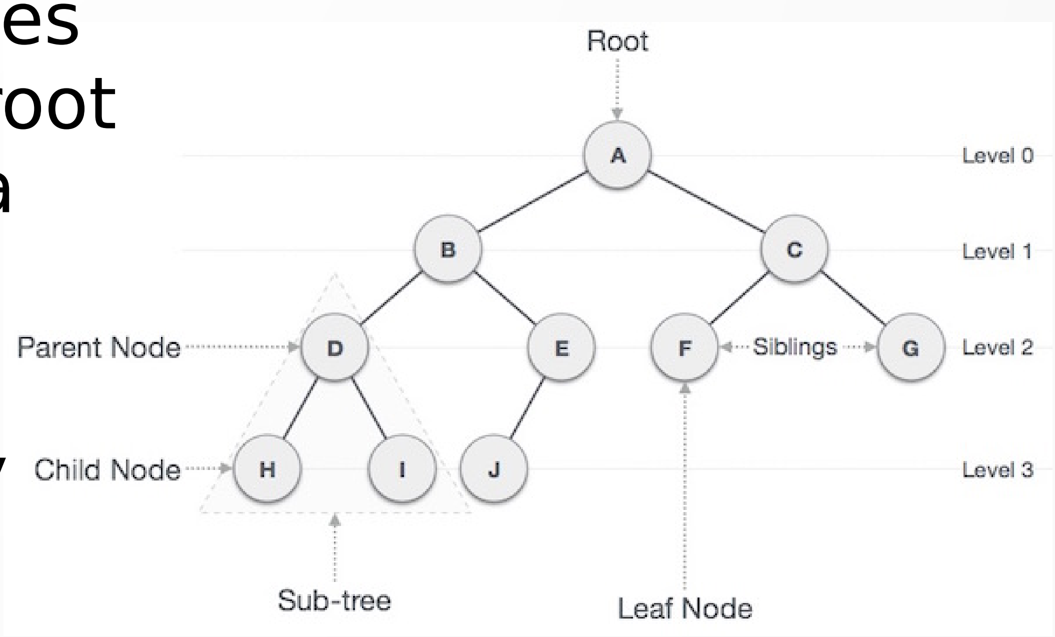




https://www.tutorialspoint.com/data_structures_algorithms/dsa_queue.htm

Overview: Trees

- In computer science, a tree is a widely used abstract data type that simulates a hierarchical tree structure, with a root value and subtrees of children with a parent node, represented as a set of linked nodes.
- We will discuss binary tree or binary search tree specifically.
- Binary Tree is a special data structure used for data storage purposes.

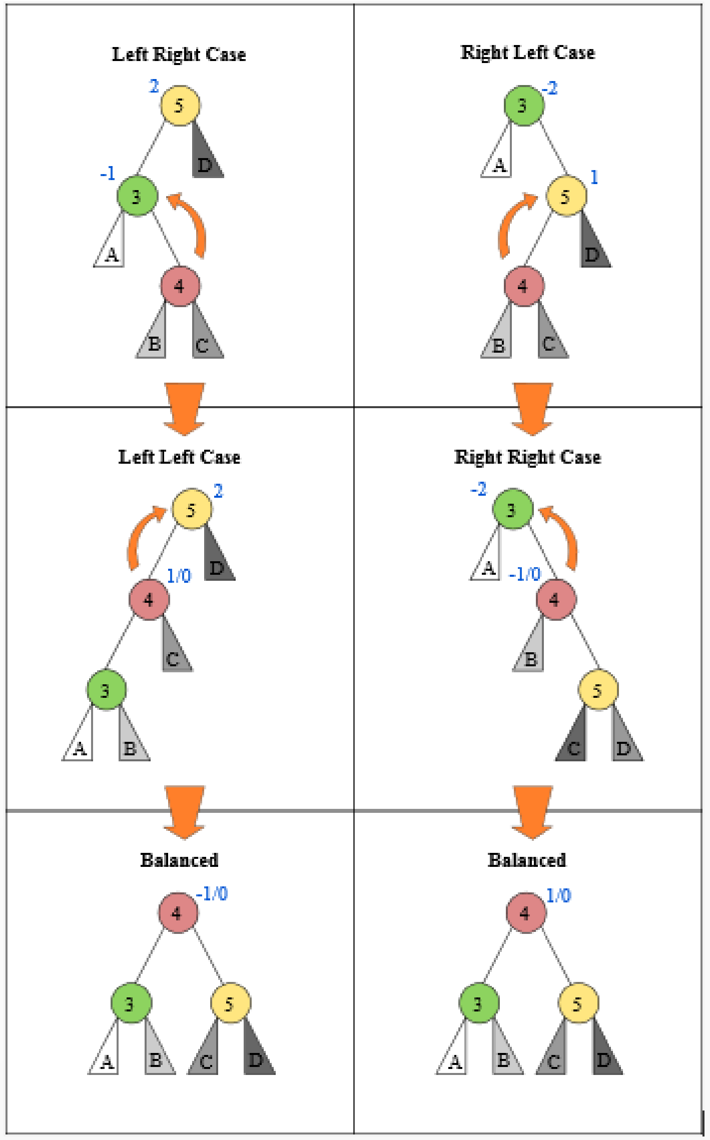


https://en.wikipedia.org/wiki/Tree_(data_structure)

https://www.tutorialspoint.com/data_structures_algorithms/tree_data_structure.htm

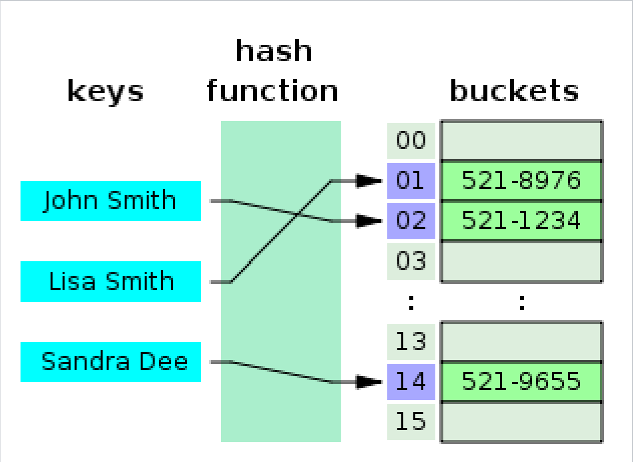
Overview: Trees

- AVL tree is a self-balancing Binary Search Tree (BST)
- The difference between heights of left and right subtrees cannot be more than one for all nodes



Overview: Hash Tables

- In computing, a hash table (hash map) is a data structure that can map keys to values.
- In many situations, hash tables turn out to be on average more efficient than search trees or any other table lookup structure.
- For this reason, they are widely used in many kinds of computer software, particularly for associative arrays (which are used for storing key value pairs), database indexing, caches, and sets.



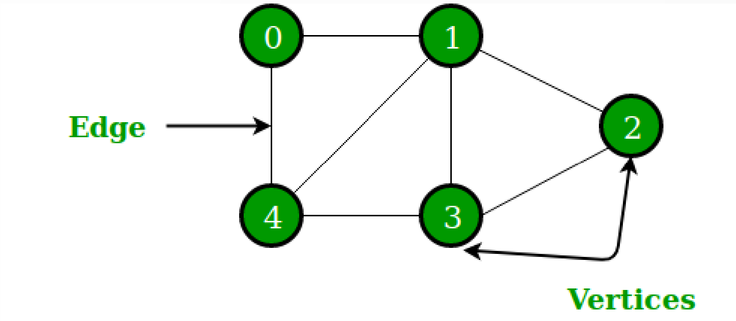
A small phone book as a hash table

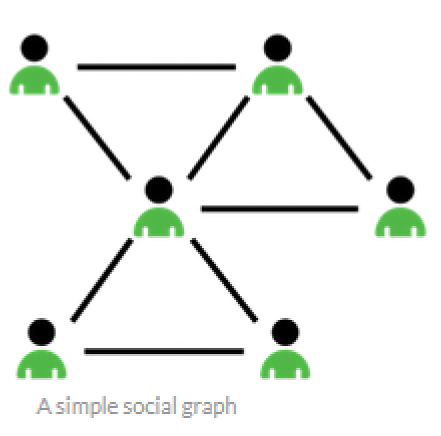
https://en.wikipedia.org/wiki/Associative_array

https://en.wikipedia.org/wiki/Hash_table

Overview: Graphs

- In computer science, a graph is an abstract data type that is meant to implement the undirected graph and directed graph concepts from the field of graph theory within mathematics.
- A graph is a pictorial representation of a set of objects where some pairs of objects are connected by links.
- The interconnected objects are represented by points termed as vertices, and the links that connect the vertices are called edges.
- A Graph is a non-linear data structure
- Graphs are used to solve many real-life problems. Graphs are used to represent networks that may include paths in a city or telephone network or circuit network.





https://www.business2community.com/branding/what-the-social-graph-is-and-why-it-matters-to-brands-0132552

https://en.wikipedia.org/wiki/Graph_(abstract_data_type))

https://www.tutorialspoint.com/data_structures_algorithms/graph_data_structure.htm

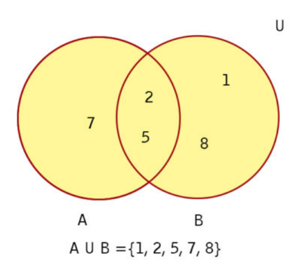
https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/

Overview: Sets

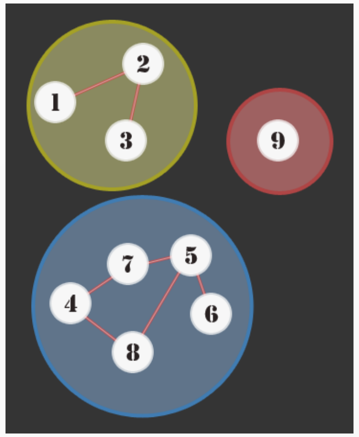
- In computer science, a set is an abstract data type that can store unique values, without any particular order.
- It is a computer implementation of the mathematical concept of a finite set.
- Sets are a type of associative containers in which each element has to be unique, because the value of the element identifies it.
- The value of the element cannot be modified once it is added to the set, though it is possible to remove and add the modified value of that element.

https://en.wikipedia.org/wiki/Set_(abstract_data_type)

https://www.geeksforgeeks.org/set-in-cpp-stl/

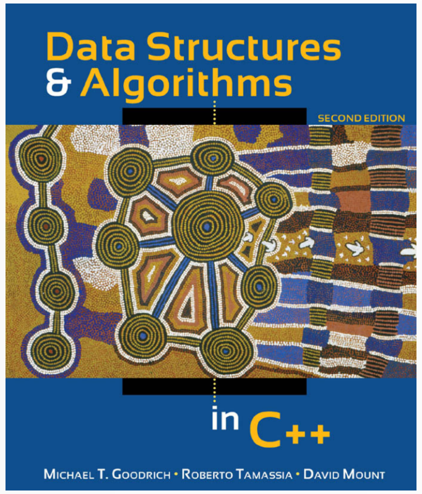


https://www.vibrantpublishers.com/2019/11/14/data-structures-for-sets/



https://www.mathblog.dk/disjoint-set-data-structure/

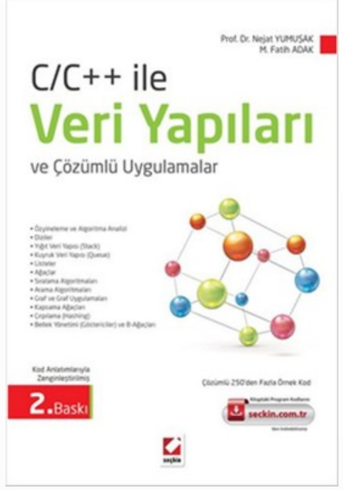
Sources


Data Structures & Algorithms

SECOND EDITION

in C++

Michael T. Goodrich • Roberto Tamassia • David Mount


Prof. Dr. Nejat YUMUŞAK
M. Fatih ADAK

C/C++ ile
Veri Yapıları
ve Çözümlü Uygulamalar

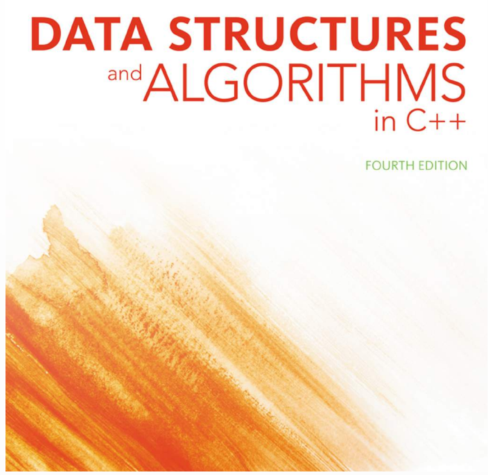
- Özyineleme ve Algoritma Analizi
- Diziler
- Yığıt Veri Yapısı (Stack)
- Kuyruk Veri Yapısı (Queue)
- Listeler
- Ağaçlar
- Sıralama Algoritmaları
- Arama Algoritmaları
- Graf ve Graf Uygulamaları
- Kapsama Ağaçları
- Çırpılama (Hashing)
- Bellek Yönetimi (Göstericiler) ve B-Ağaçları

Kod Anlatımlarıyla Zenginleştirilmiş

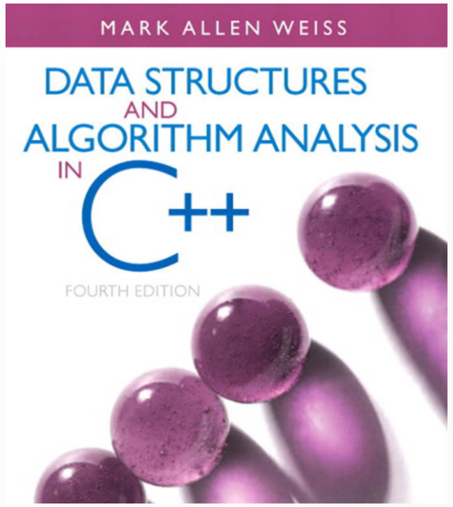
2. Baskı

Çözümlü 250'den Fazla Örnek Kod

seckin.com.tr

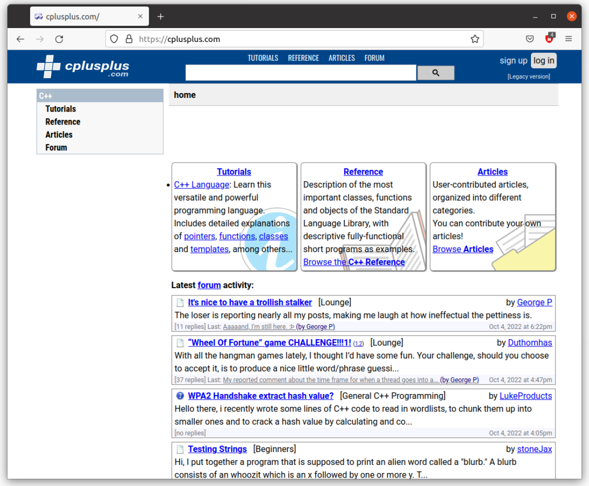

DATA STRUCTURES and ALGORITHMS in C++

FOURTH EDITION


MARK ALLEN WEISS

DATA STRUCTURES AND ALGORITHM ANALYSIS IN C++

FOURTH EDITION



Tutorials

- C++ Language: Learn this versatile and powerful programming language. Includes detailed explanations of pointers, functions, classes and templates, among others...

Reference

Description of the most important classes, functions and objects of the Standard Language Library, with descriptive fully-functional short programs as examples.
Browse the C++ Reference

Articles

User-contributed articles, organized into different categories.
You can contribute your own articles!
Browse Articles

Latest forum activity:

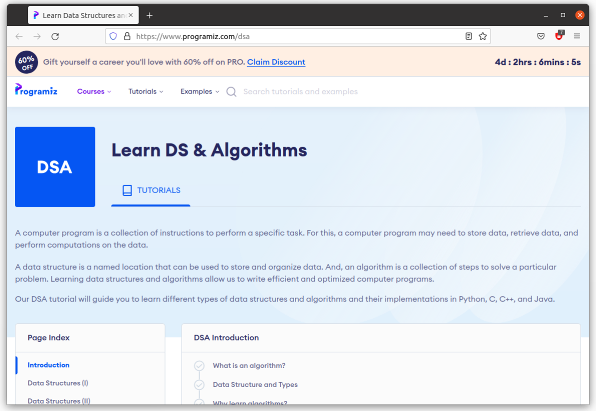
It's nice to have a trollish stalker [Lounge] by George P
The loser is reporting nearly all my posts, making me laugh at how ineffectual the pettiness is.
[11 replies] Last: Aaaaand, I'm still here. :Þ (by George P) Oct 4, 2022 at 6:22pm

"Wheel Of Fortune" game CHALLENGE!!!1! (1,2) [Lounge] by Duthomhas
With all the hangman games lately, I thought I'd have some fun. Your challenge, should you choose to accept it, is to produce a nice little word/phrase guessi...
[37 replies] Last: My reported comment about the time frame for when a thread goes into a... (by George P) Oct 4, 2022 at 4:47pm

WPA2 Handshake extract hash value? [General C++ Programming] by LukeProducts
Hello there, i recently wrote some lines of C++ code to read in wordlists, to chunk them up into smaller ones and to crack a hash value by calculating and co...
[no replies] Oct 4, 2022 at 4:05pm

Testing Strings [Beginners] by stoneJax
Hi, I put together a program that is supposed to print an alien word called a "blurb." A blurb consists of an whoozit which is an x followed by one or more y. T...

https://cplusplus.com/



Gift yourself a career you'll love with 60% off on PRO. Claim Discount 4d : 2hrs : 6mins : 5s

Learn DS & Algorithms

A computer program is a collection of instructions to perform a specific task. For this, a computer program may need to store data, retrieve data, and perform computations on the data.

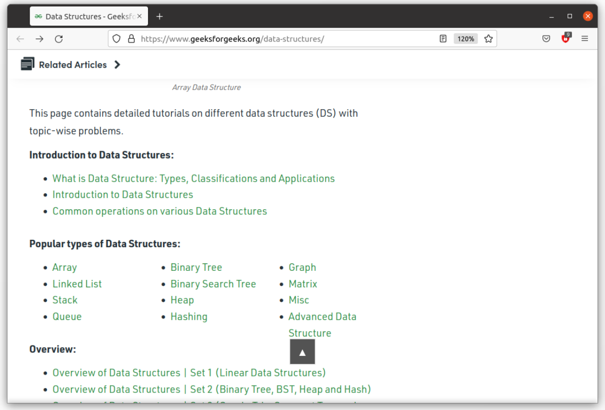
A data structure is a named location that can be used to store and organize data. And, an algorithm is a collection of steps to solve a particular problem. Learning data structures and algorithms allow us to write efficient and optimized computer programs.

Our DSA tutorial will guide you to learn different types of data structures and algorithms and their implementations in Python, C, C++, and Java.

Page Index
- Introduction
- Data Structures (I)
- Data Structures (II)

DSA Introduction
- What is an algorithm?
- Data Structure and Types
- Why learn algorithms?

https://www.programiz.com/dsa


Data Structures - Geeksfo
https://www.geeksforgeeks.org/data-structures/
120%
Related Articles

Array Data Structure

This page contains detailed tutorials on different data structures (DS) with topic-wise problems.

Introduction to Data Structures:

- What is Data Structure: Types, Classifications and Applications
- Introduction to Data Structures
- Common operations on various Data Structures

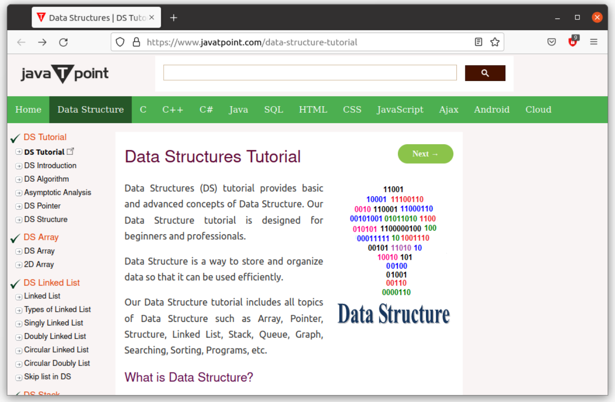
Popular types of Data Structures:

- Array
- Linked List
- Stack
- Queue
- Binary Tree
- Binary Search Tree
- Heap
- Hashing
- Graph
- Matrix
- Misc
- Advanced Data Structure

Overview:

- Overview of Data Structures | Set 1 (Linear Data Structures)
- Overview of Data Structures | Set 2 (Binary Tree, BST, Heap and Hash)

https://www.geeksforgeeks.org/data-structures/



Data Structures Tutorial

Data Structures (DS) tutorial provides basic and advanced concepts of Data Structure. Our Data Structure tutorial is designed for beginners and professionals.

Data Structure is a way to store and organize data so that it can be used efficiently.

Our Data Structure tutorial includes all topics of Data Structure such as Array, Pointer, Structure, Linked List, Stack, Queue, Graph, Searching, Sorting, Programs, etc.

What is Data Structure?

https://www.javatpoint.com/data-structure-tutorial