CS2400 - Data Structures and Advanced Programming
Module 1: Introduction

Hao Ji
Computer Science Department
Cal Poly Pomona

Data Structures

- One of the most fundamental topics in Computer Science
  - A particular way to store and organize **data** so that it can be used **efficiently**.

A good data structure can make massive savings in computation time and space!

Data Structures

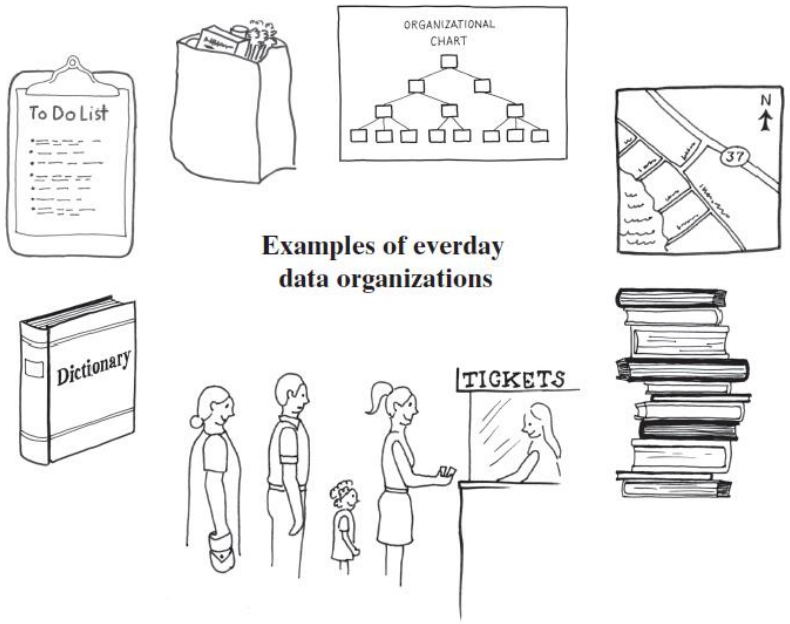
- One of the most fundamental topics in Computer Science
  - A particular way to store and organize data so that it can be used efficiently.

Almost every high-tech company asks data structure questions from this course in coding interviews.

Data Structures

- One of the most fundamental topics in Computer Science
  - A particular way to store and organize data so that it can be used efficiently.

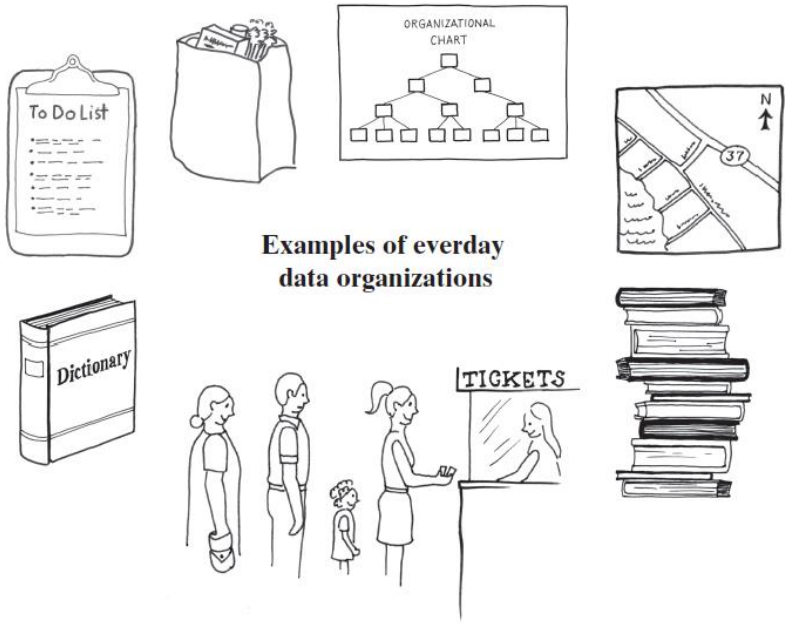
- (Look around and see) Data organization in life
  - Standing in a line
  - Stack of books
  - To-Do list
  - Dictionary
  - Road map
  - …



Examples of everday data organizations

Data Structures (Cont'd)

- **Collection** contains a group of objects

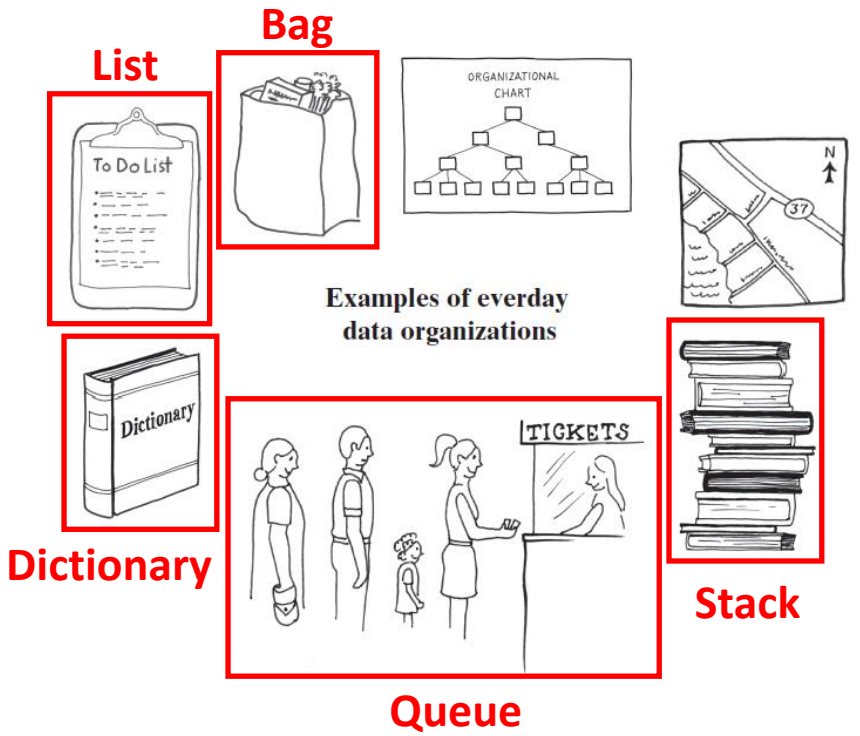


Examples of everday data organizations

Data Structures (Cont'd)

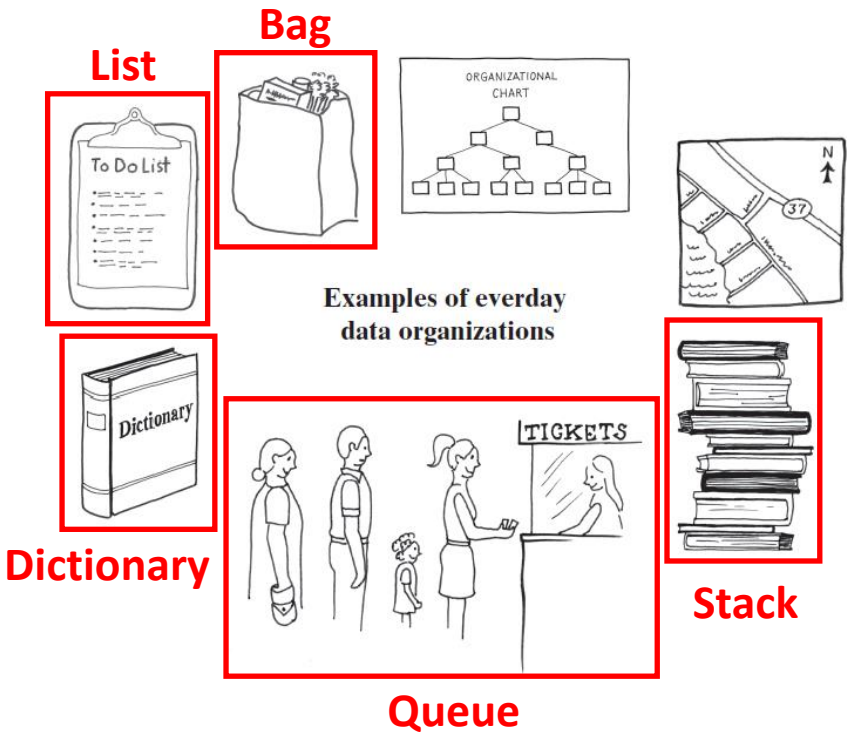
- **Collection** contains a group of objects

- Bag, List, Queue, Stack, and Dictionary
  - Each of them is a collection that stores its entries in a linear sequence, and in which entries may be added or removed at will.
  - They differ in the restrictions they place on how these entries may be added, removed, or accessed.



Data Structures (Cont'd)

- Bag consists of an unordered collection that allows duplicates.
- List numbers its items.
- Queue and Stack order their items chronologically.
- Dictionary contains pairs of items.

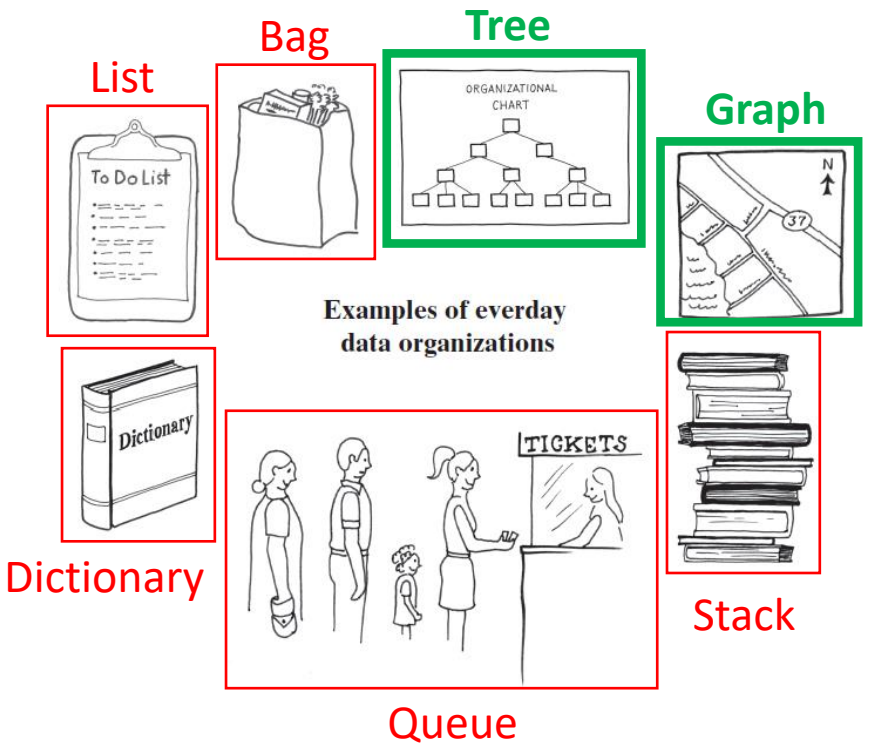


Data Structures (Cont'd)

- **Collection** contains a group of objects

- Bag, List, Queue, Stack, and Dictionary
  - Each of them is a collection that stores its entries in a linear sequence, and in which entries may be added or removed at will.
  - They differ in the restrictions they place on how these entries may be added, removed, or accessed.

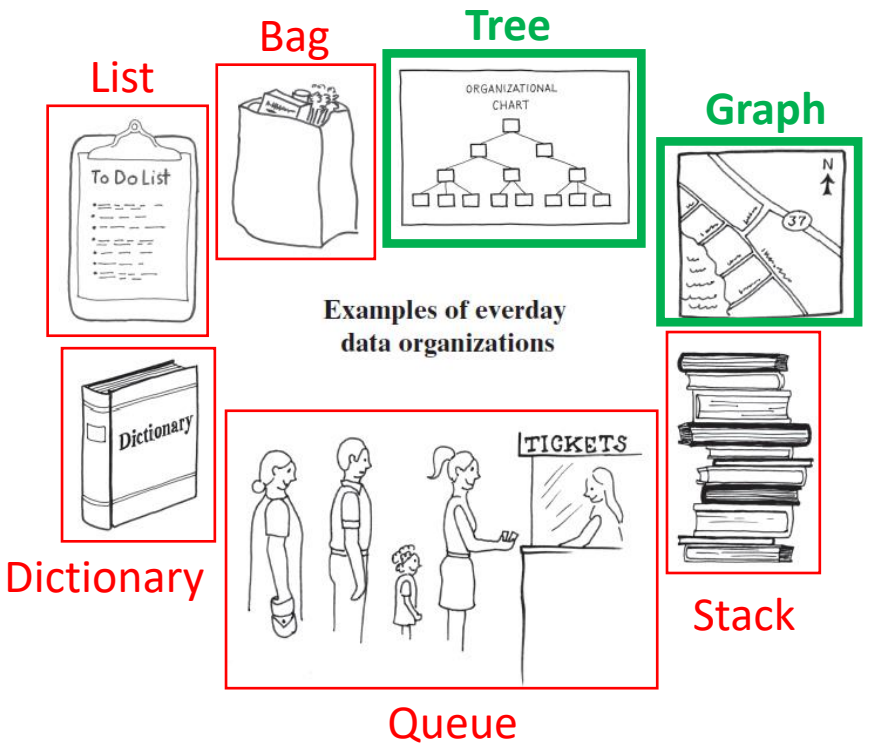
- Tree and Graph
  - Data entries are not arranged in a sequence, but with different rules



Data Structures (Cont'd)

- Tree organizes its entries according to some hierarchy.

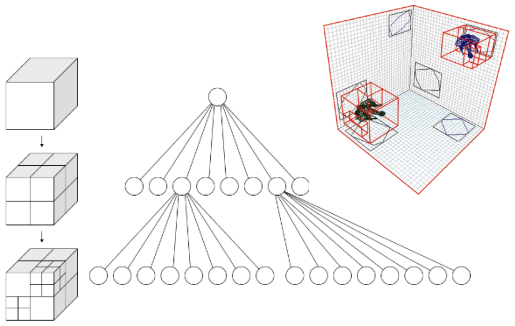
- Graph is a generalization of the ADT tree that focuses on the relationship among its entries instead of any hierarchical organization.

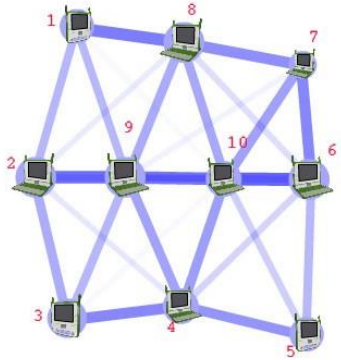


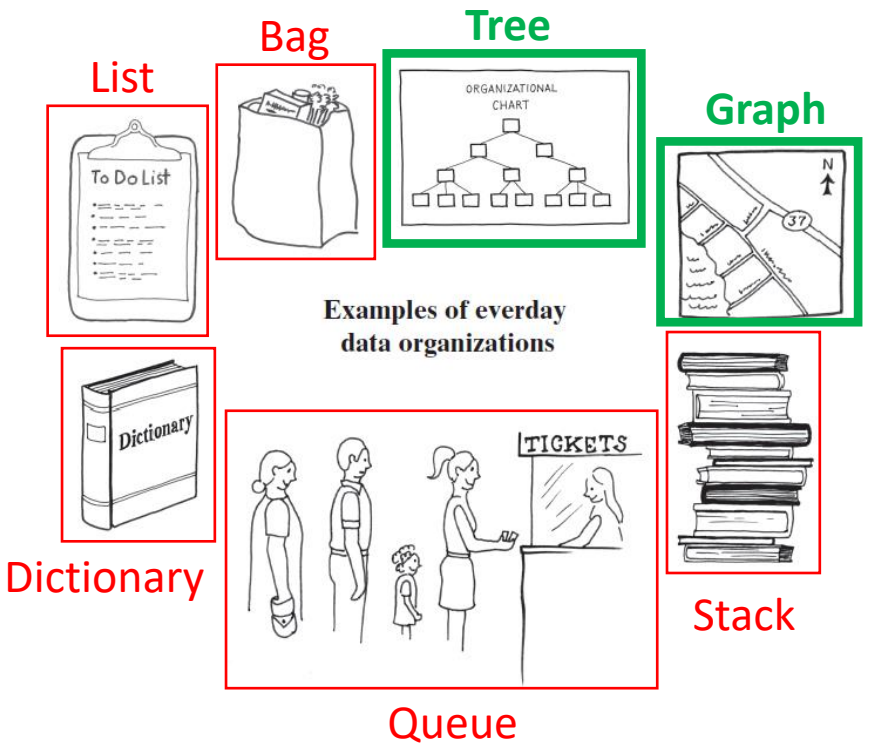
Data Structures (Cont'd)

- Tree organizes its entries according to some hierarchy.

- Graph is a generalization of the ADT tree that focuses on the relationship among its entries instead of any hierarchical organization.







Data Structures (Cont'd)

- To represent real world objects into computer programs, we need to consider
  - What data we want to store?
  - What operations we want on the data?
  - How to store the data?
  - How to implement the operations?

Data Structures (Cont'd)

- To represent real world objects into computer programs, we need to consider
  - What data we want to store?
  - What operations we want on the data?
  - How to store the data?
  - How to implement the operations?

ADT

Data Structures (Cont'd)

- To represent real world objects into computer programs, we need to consider
  - What data we want to store?
  - What operations we want on the data?
  - How to store the data?
  - How to implement the operations?

ADT

An abstract data type, or ADT, is a specification that describes a data set and the operations on that data.

Data Structures (Cont'd)

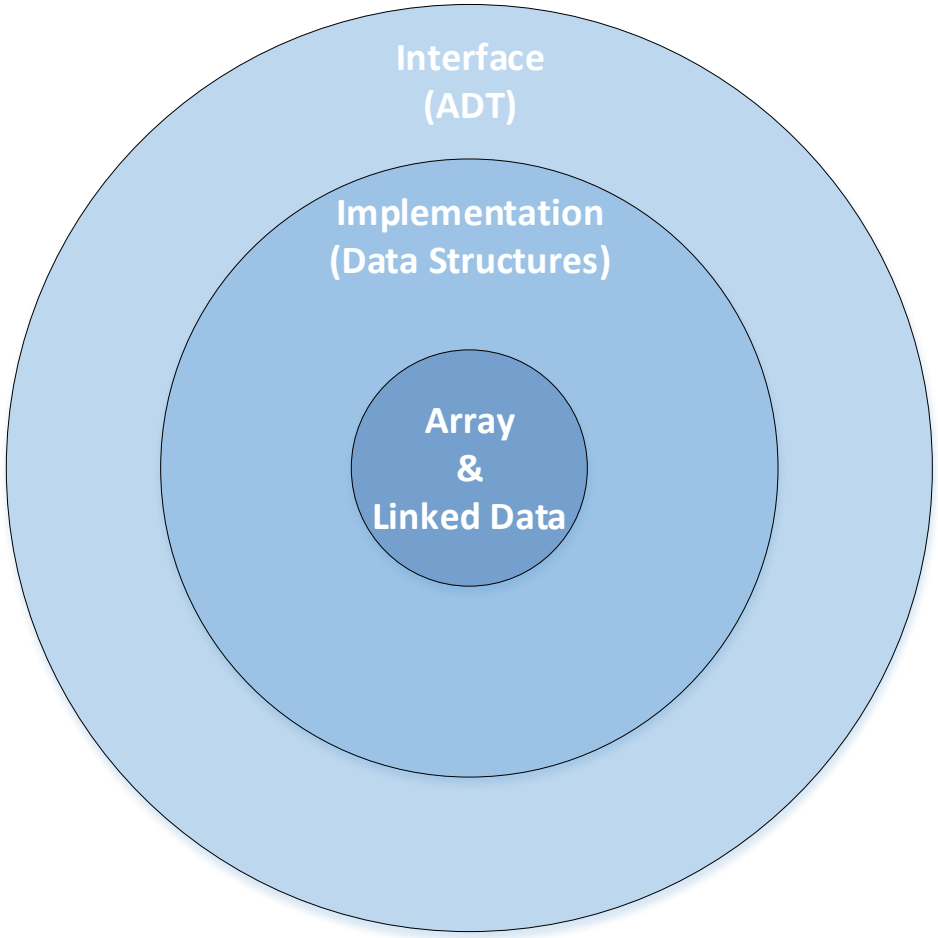
- To represent real world objects into computer programs, we need to consider
  - What data we want to store?
  - What operations we want on the data?
  - How to store the data?
  - How to implement the operations?

Data Structures

Data structures is an implementation of an ADT with a programming language, which focuses on the concrete issues of implementations.

Data Structures (Cont'd)

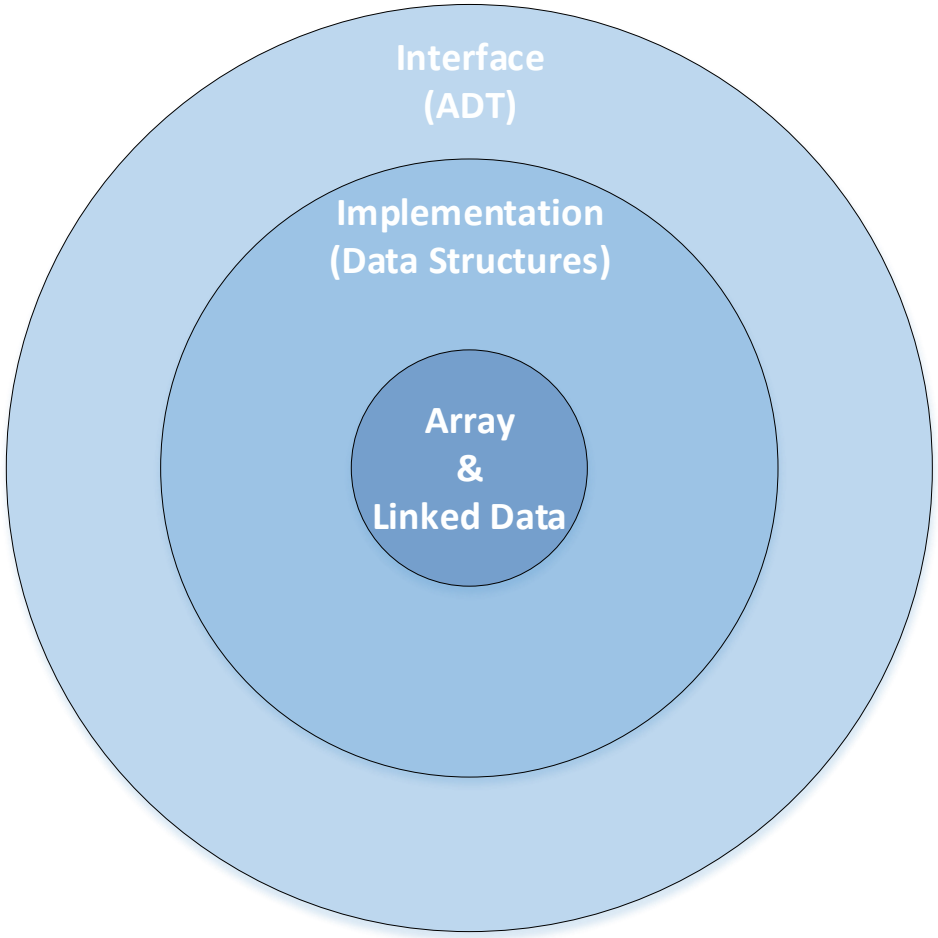
- An ADT makes a clean separation between interface and implementation
  - The user only sees the interface and therefore does not need to tamper with the implementation.
  - The abstraction makes the code more robust and easier to maintain.



Data Structures (Cont'd)

- An ADT makes a clean separation between interface and implementation
  - The user only sees the interface and therefore does not need to tamper with the implementation.
  - The abstraction makes the code more robust and easier to maintain.

- We will study different abstract data types, different data structures, their java implementations, and efficiency analysis.



What Is an Algorithm?

- An algorithm is a procedure
  - A finite set of well-defined instructions, for solving a problem which, given an initial state, will terminate in a defined end-state.

- The computational complexity and efficient implementation of the algorithm are important in computing, and this depends on suitable data structures.

- Efficiency of algorithms
  - Running Time (Time complexity)
  - Memory Cost (Space complexity)

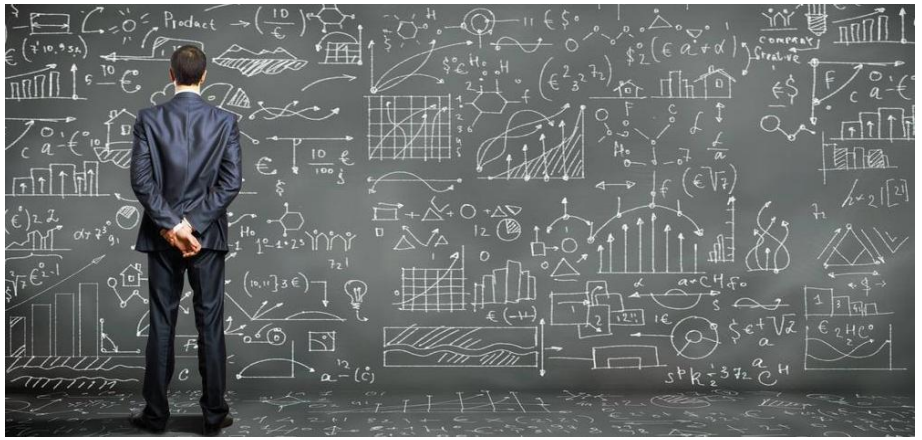
How Do I Choose the Right Data Structures?

- When writing a program, one of the first steps is determining or choosing the data structures.

- What are the "right" data structures for the program?
  - The interface of operations supported by a data structure is one factor to consider when choosing between several available data structures.
  - Another important factor is the efficiency of the data structure: how much space does the data structure occupy, and what are the running times of the operations in its interface?

Moving Algorithm into Development

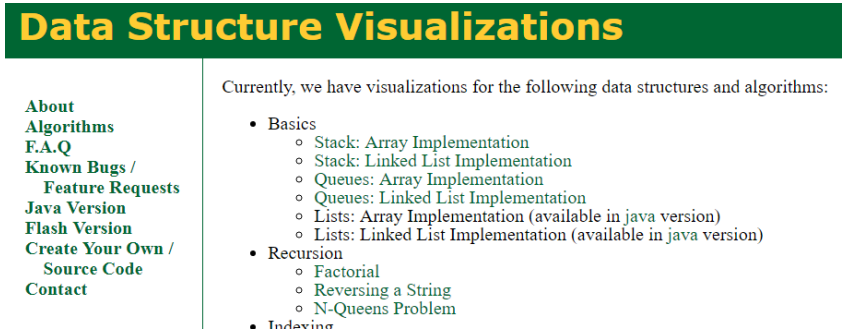
- The Development Process (not necessarily in order):
  - Specification of the task
  - Design of a solution
  - Implementation (coding) of the solution
  - Analysis of the solution
  - Testing and debugging
  - Maintenance



http://www.tarikguney.com/do-not-ignore-algorithms-and-data-structures/

Interactive and Visualization Tools

- https://www.cs.usfca.edu/~galles/visualization/Algorithms.html



- https://visualgo.net/en

Data Structures and **Advanced Programming**

Java

Please see
“Module 2 - Review of Java Programming Basics, Interface, and Generic Data Types.pdf”