CSE-281: Data Structures and Algorithms

Introduction

Ref. Book: Schaum's Outline Series, Theory and problems of Data Structures
By Seymour Lipschutz

Eftekhar Hossain
Assistant Professor
Dept. of ETE, CUET

Course Outline

- Concepts and Examples
- Elementary Data Objects
- Elementary Data Structures
- Arrays
- Lists
- Stacks
- Queues
- Graphs
- Trees
- Sorting and Searching
- Hash Techniques

Books

- Schaum's Outline Series, Theory and problems of Data Structures

  By Seymour Lipschutz

- Data Structures and Algorithms

  By A. V. Aho, J. E. Hopcroft, J. D. Ullman

- Data Structures Using C and C++

  By Y. Langsam, M. J. Augenstein, A. M. Tenenbaum

- Fundamentals of Computer Algorithms

  By Ellis Horowitz, Sartaz Sahni

Introduction

- To exactly know, what is data structure? We must know:
  - ***What is a computer program?***

Figure 1: Input-Processing-Output

Elementary Data Organization

- Data are simply values or sets of values.
- Collection of data are frequently organized into a hierarchy of fields, records and files.
- This organization of data may not complex enough to maintain and efficiently process certain collections of data.
- For this reason, data are organized into more complex type of structures called Data Structure.

Elementary Data Organization

- 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.

Data Structure

- Definition — In computer science, a data structure is a data organization, management and storage format that enables efficient access and modification.
- In Simple Words —

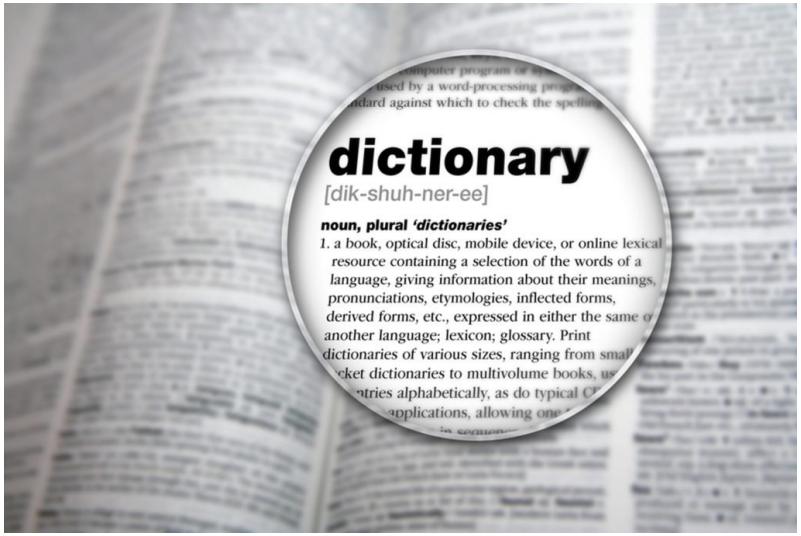
Data Structure is a way in which data is stored on a computer.

Why do we need Data Structure

Data structure is a particular way of storing and organizing information in a computer so that it can be retrieved and used most productively.

- Each Data Structure allows data to be stored in specific manner.
- Data Structure allows efficient data search and retrieval.
- Specific Data structures are decided to work for specific problems.
- It allows to manage large amount of data such as large databases and indexing services such as hash table.

Real World Scenario



Data Structures

Data Structures

The logical or mathematical model of a particular organization of data is called a data structure.

Types of Data Structure

1. Linear Data Structure
   Example: Arrays, Linked Lists, Stacks, Queues
2. Nonlinear Data Structure
   Example: Trees, Graphs

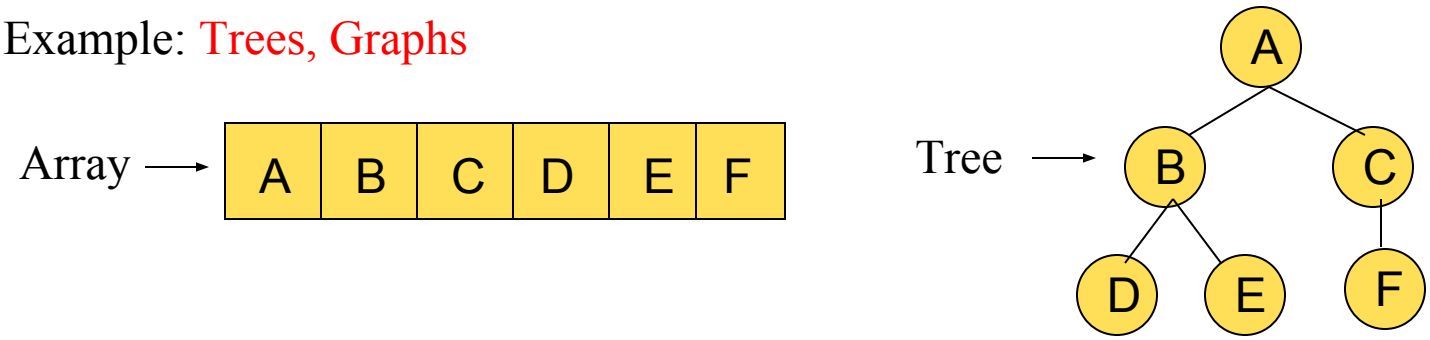


Figure 2: Linear and nonlinear structures

Which data structure to use?

- Data structures let the input and output be represented in a way that can be handled efficiently and effectively.

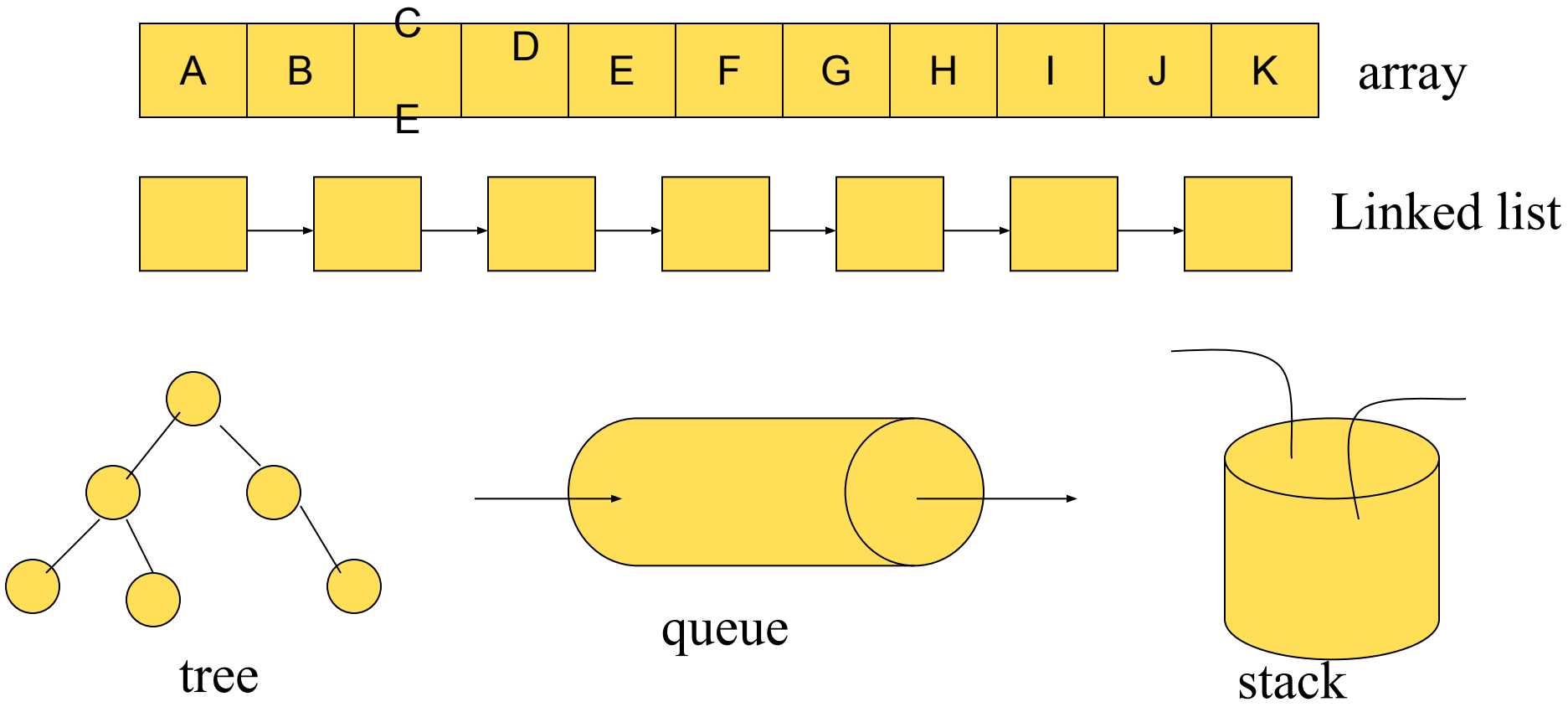


Figure 3: Different Data Structures

Data Structures

- **3 steps in the study of data structures**
  - Logical or mathematical description of the structure
  - Implementation of the structure on the computer
  - Quantitative analysis of the structure, which includes determining the amount of memory needed to store the structure and the time required to process the structure

Choice of Data Structures

The choice of data structures depends on two considerations:

1. It must be rich enough in structure to mirror the actual relationships of data in the real world.
2. The structure should be simple enough that one can effectively process data when necessary.

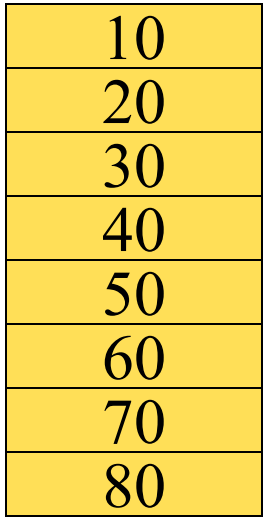


Figure 4: Array with 8 items

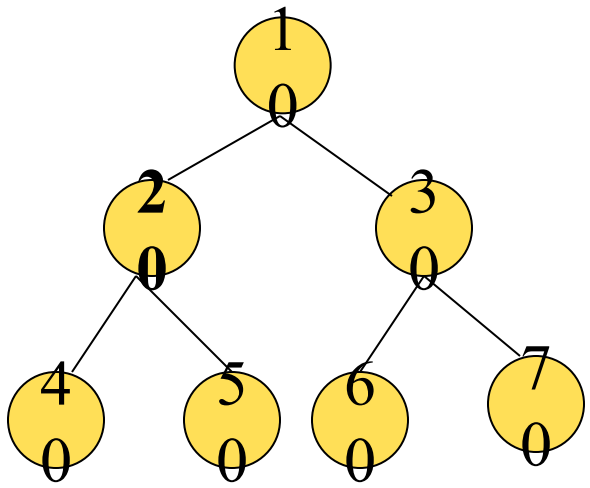


Figure 5: Tree with 8 nodes

Data Structure Operations

1. **Traversing:** Accessing each record exactly once so that certain items in the record may be processed.
2. **Searching:** Finding the location of the record with a given key value.
3. **Inserting:** Adding a new record to the structure.
4. **Deleting:** Removing a record from the structure.
5. **Sorting:** Arranging the records in some logical order.
6. **Merging:** Combing the records in two different sorted files into a single sorted file.

Thank You

CSE-281: Data Structures and Algorithms

Classification of Data Structure

Ref: *Online Resource*

Eftekhar Hossain
Assistant Professor
Dept. of ETE, CUET

Data Structure

- Data Structure is a way of collecting and organizing data in such a way that we can perform operations on these data in an effective way...

- Searching data.
- Need to manage processor speed.
- Serve multiple request simultaneously.

Characteristics of a Data Structure

- Time Complexity - Running time or the execution time of operations of data structure must be as small as possible.
- Space Complexity - Memory usage of a data structure operation should be as little as possible..

Execution Time Cases

There are three cases which are usually used to compare various data structures execution time :

- Worst Case — when a particular data structure operation takes maximum time.
- Average Case — This is the scenario depicting the average execution time of an operation of a data structure.
- Best Case — This is the scenario depicting the least possible execution time of an operation of a data structure.

Types Of Data Structures

- Primitive data structures
- Non-primitive data structure
  - Linear DS
  - Non Linear DS
- **Primitive** Data Structures are the basic data structures that directly operate upon the machine instructions.

  Example: int , float, char, and pointer

Types Of Data Structures

Non-primitive Data Structures

- are more complicated data structures and are derived from primitive data structures.
- emphasize on grouping same or different data items with relationship between each data item.
- Example : Array, List

Types Of Data Structures

Linear DS:

- every item is related to its previous and next time.
- data is arranged in linear sequence.
- data items can be traversed in a single run.
- implementation is easy

Example: Stack , Queue

Types Of Data Structures

Non-linear DS:

- every item is attached with many other items.
- data is not arranged in sequence.
- data cannot be traversed in a single run.
- implementation is difficult.

Example: Tree, Graph

Static and Dynamic DS

Static

- Static data structures are those whose sizes and structures associated memory locations are fixed at compile time. Example: Array

Dynamic

- Dynamic structures are those which expands or shrinks depending upon the program need and its execution. Also, their associated memory locations changes. Example: Linked List created using pointers

Cont.

Homogeneous

- In homogeneous data structures, all the elements are of same type.

  Example: Array

Non-Homogeneous

- In Non-Homogeneous data structure, the elements may or may not be of the same type.

  Example: Structures

What is an Algorithm?

- A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

Formal Definition :

- An algorithm is a finite set of instructions that are carried in a specific order to perform specific task.

Algorithm Characteristics

- Algorithms typically have the following characteristics
- Inputs : 0 or more input values.
- Outputs : 1 or more than 1 output.
- Unambiguity : clear and simple instructions.
- Finiteness : Limited number of instructions.
- Effectiveness : Each instruction has an impact on the overall process.

Thank You

CSE-281: Data Structures and Algorithms

Data Structures
(Chapter-1)

Ref: Schaum's Outline Series, Theory and problems of Data Structures
By Seymour Lipschutz

Eftekhar Hossain
Assistant Professor
Dept. of ETE, CUET

Arrays

- The simplest type of data structure is a linear (or one-dimensional) array.
- list of a finite number n of similar data elements referenced respectively by a set of n consecutive numbers, usually 1, 2, 3, ..., n.
- choose the name A for the array, then the elements of A are denoted by bracket notation

  A[1], A[2], A[3]….. A[N]

- the number K in A[K] is called a subscript and A[K] is called a subscripted variable.

*

Example

- A linear array STUDENT consisting of the names of six students is pictured in Table. Here STUDENT[1] denotes John Brown, STUDENT[2] denotes Sandra Gold, and so on.

	STUDENT
0	Jhon Brown
1	Sandra Gold
2	Akbar
3	AB De
4	Alan Smith
5	Tom Jones

*

Example

- Linear arrays are called one-dimensional arrays because each element in such an array is referenced by one subscript.

- A two-dimensional array is a collection of similar data elements where each element is referenced by two subscripts.

- Example 2
- A chain of 28 stores, each store having 4 departments, may list its weekly sales (to the nearest dollar)., then

*

Example

Store \ Dept.	1	2	3	4
1	2872	805	3211	1560
2	2196	1223	2525	1744
3	3257	1017	3686	1951
. . .	. . .	. . .	. . .	. . .
28	2618	931	2333	982

SALES[1, 1] = 2872,
SALES[1, 2] = 805,
SALES[1, 3] = 3211,
SALES[28, 4] = 982

*

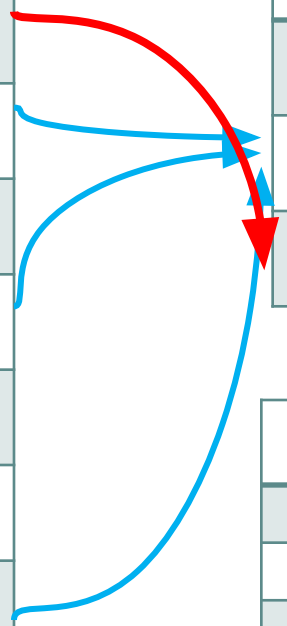
Linked Lists

	Customer	Salesperson
1	Adams	Smith
2	Brown	Ray
3	Clark	Jones
4	Drew	Ray
5	Evans	Smith
6	Farmer	Jones
7	Geller	Ray
8	Hill	Smith
9	Infeld	Ray

*

Linked Lists

	Customer	Pointer
1	Adams	3
2	Brown	2
3	Clark	1
4	Drew	2
5	Evans	3
6	Farmer	1
7	Geller	2
8	Hill	3
9	Infeld	2



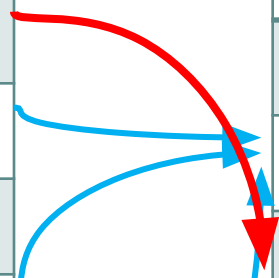
Salesperson	
Jones	1
Ray	2
Smith	3

	Customer	Salesperson
1	Adams	Smith
2	Brown	Ray
3	Clark	Jones
4	Drew	Ray
5	Evans	Smith
6	Farmer	Jones
7	Geller	Ray
8	Hill	Smith
9	Infeld	Ray

*

Linked Lists

	Customer	Pointer
1	Adams	3
2	Brown	2
3	Clark	1
4	Drew	2
5	Evans	3
6	Farmer	1
7	Geller	2
8	Hill	3
9	Infeld	2



Salesperson	
Jones	1
Ray	2
Smith	3

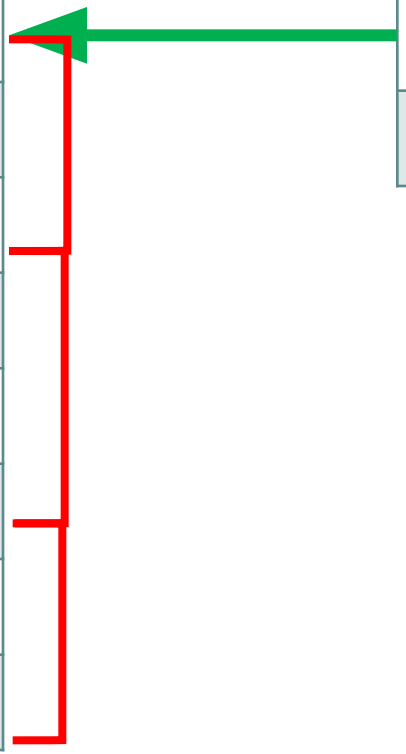
Salesperson	pointer
Jones	3, 6
Ray	2, 4, 7, 9
Smith	1, 5, 8

*

Linked Lists

	Customer	Link
1	Adams	5
2	Brown	4
3	Clark	6
4	Drew	7
5	Evans	8
6	Farmer	0
7	Geller	9
8	Hill	0
9	Infeld	0

Salesperson	pointer
Jones	3
Ray	2
Smith	1



*

Linked Lists

- Although the terms "pointer" and "link" are usually used synonymously,
- we will try to use the term "pointer" when an element in one list points to an element in a different list,
- and to reserve the term "link" for the case when an element in a list points to an element in the same list

*

Trees

- Data frequently contain a hierarchical relationship between various elements.
- The data structure which reflects this relationship is called a rooted tree graph or, simply, a tree.

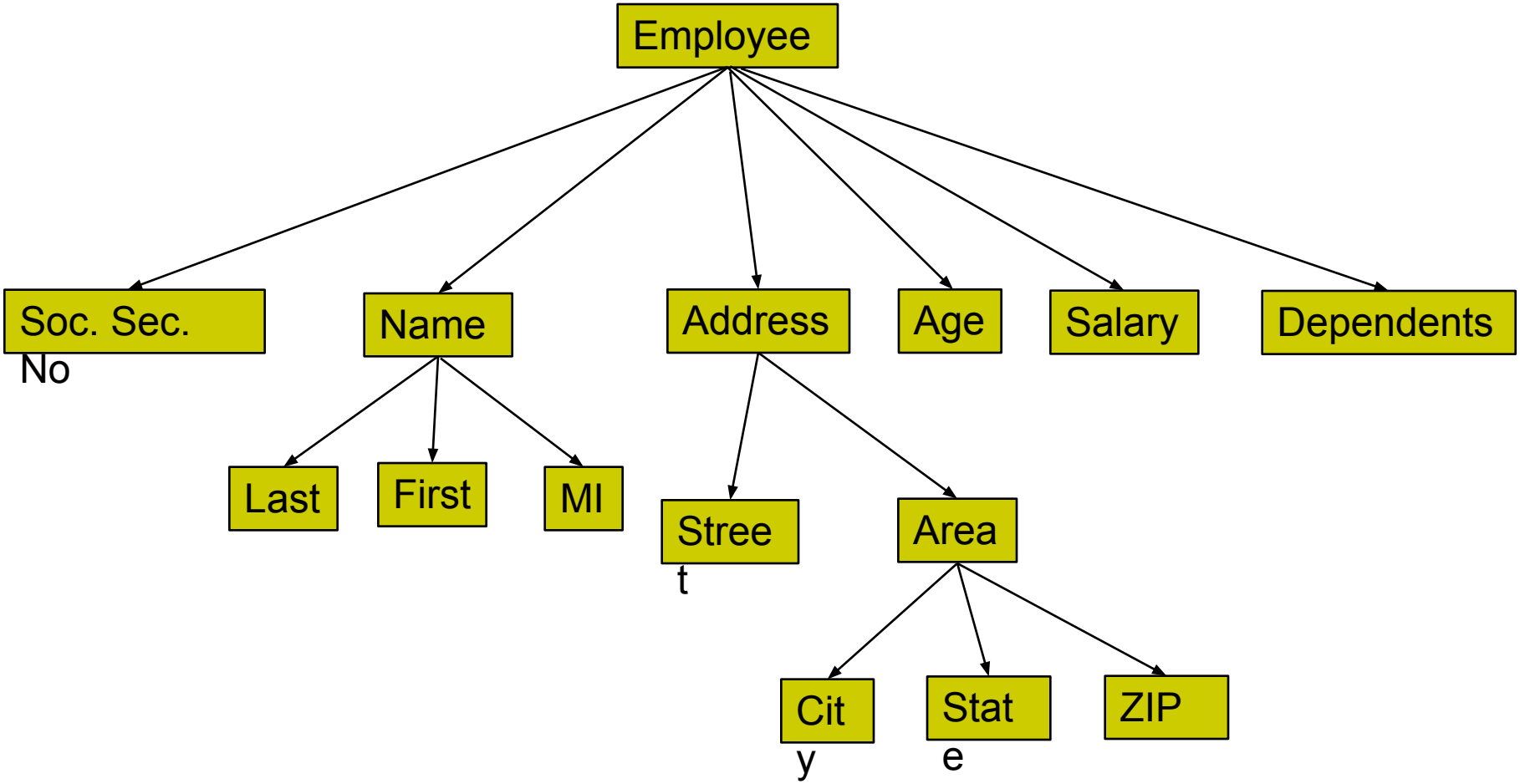
- For example, an employee personnel record may contain the following data items:
- Social Security Number, Name, Address, Age, Salary, Dependents

.

*

Trees

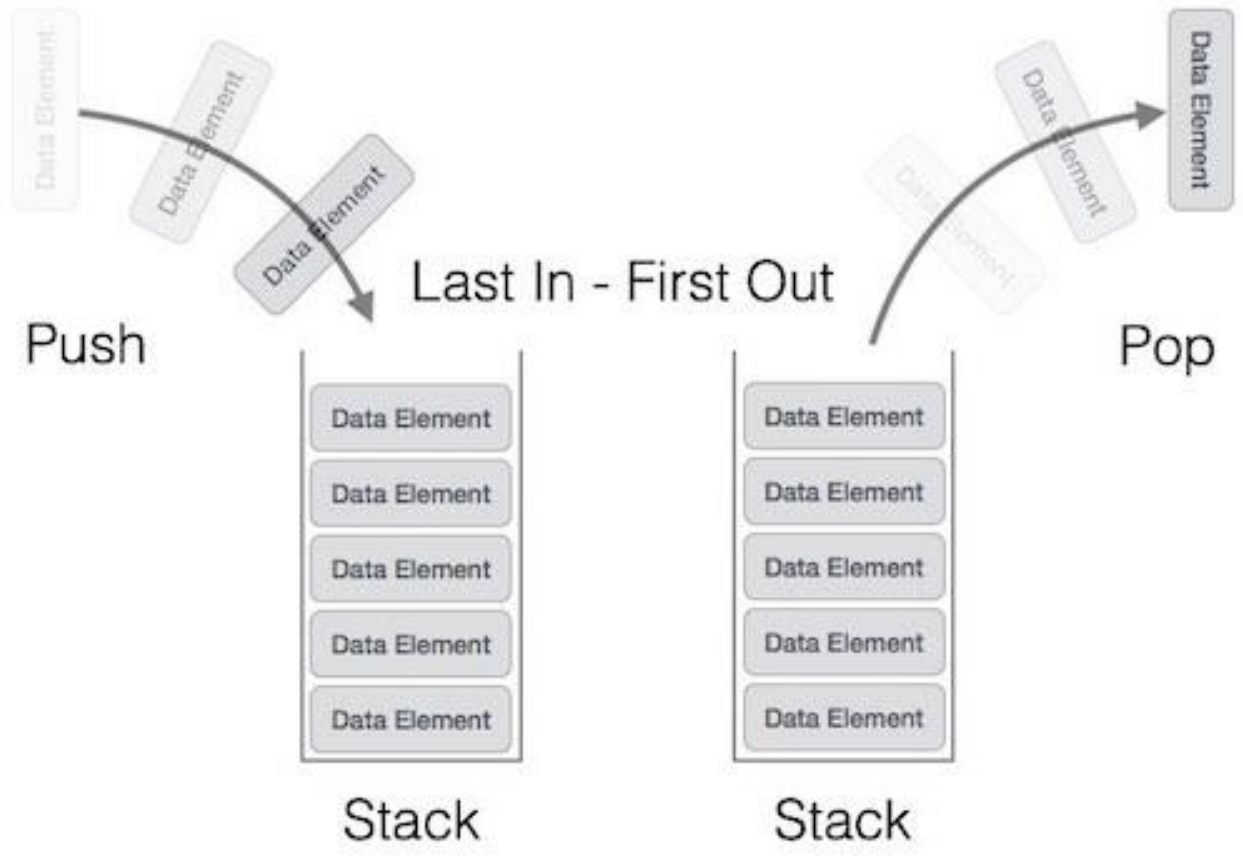
- One way of tree structure



*

Stack

- A stack, also called a last-in first-out (LIFO) system, is a linear list in which insertions and deletions can take place only at one end, called the top.



*

Queue

- A queue, also called a first in first out (FIFO) system, is a linear list in which deletions can take place only at one end of the list,
- the "front" of the list, and insertions can take place only at the other end of the list, the "rear" of the list.
- This structure operates in much the same way as a line of people waiting at a bus stop,

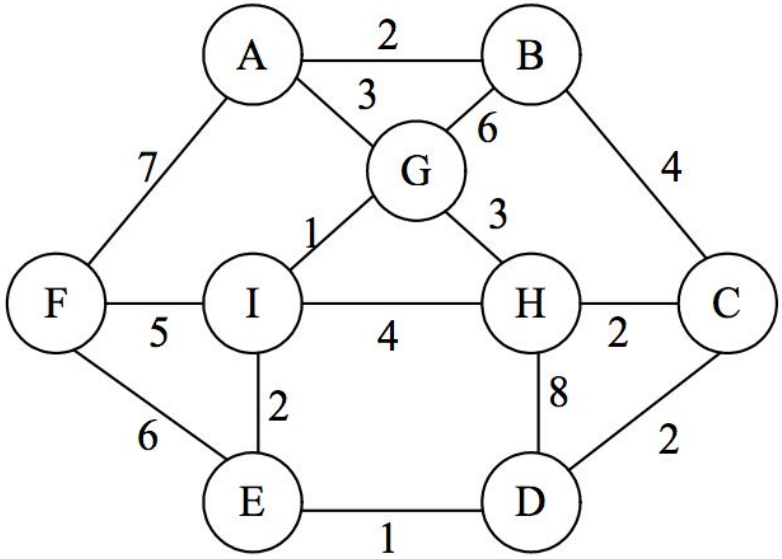
*

Example

*

Graph

- Data sometimes contain a relationship between pairs of elements which is not necessarily hierarchical in nature.
- For example. suppose an airline flies only between the cities connected by lines



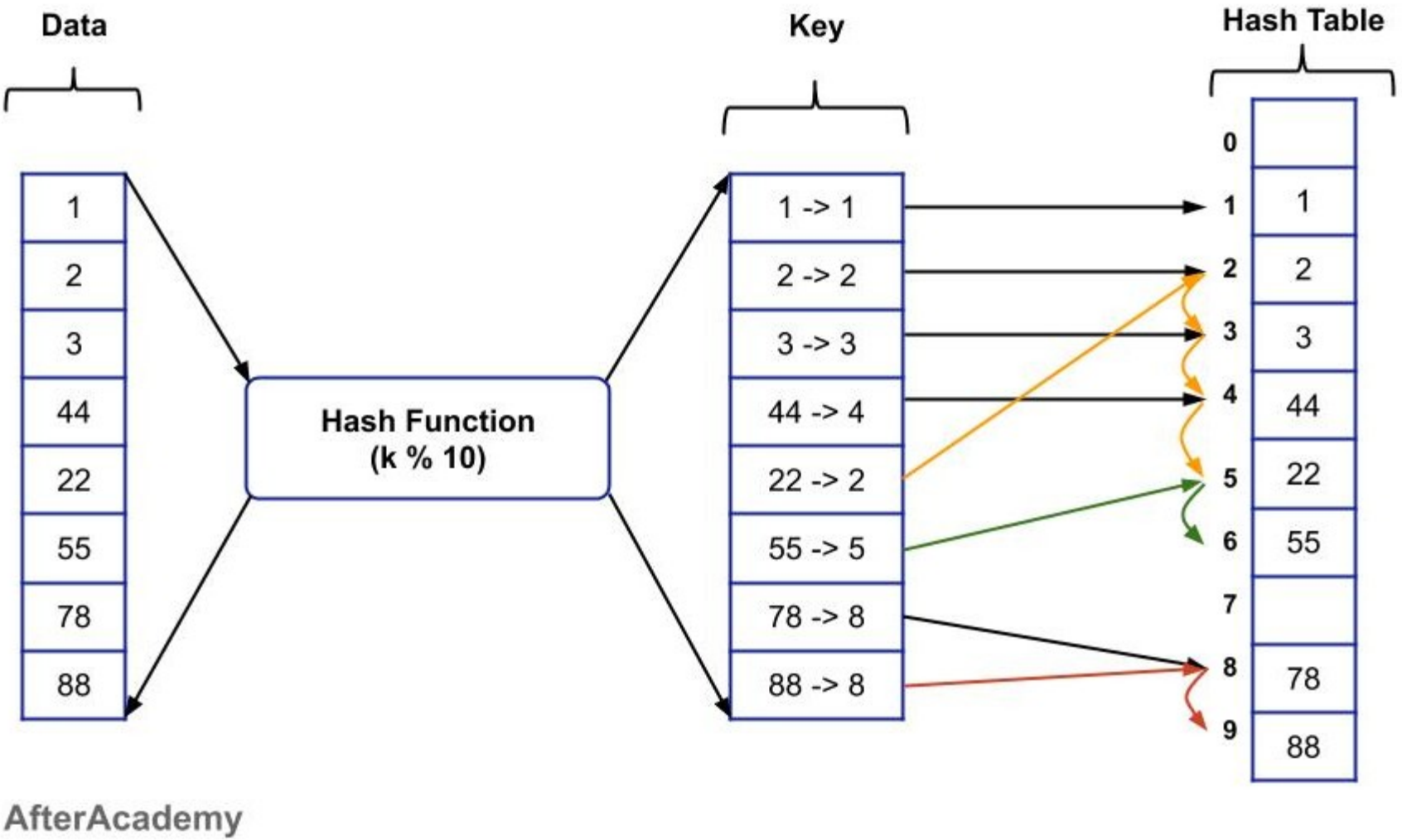
*

Hashing

- Hashing is a way to store data into some data structure (generally Hash Table is used) in such a way that
- the basic operations on that data i.e. the insertion, deletion, and searching can be performed with constant time.
- Here data is stored in the form of key-value pairs i.e. for each data you will assign some key
- and based on that key the insertion, deletion, and searching of your data will be performed.

*

Hashing


Data
Key
Hash Table
1
2
3
44
22
55
78
88
Hash Function
(k % 10)
1 -> 1
2 -> 2
3 -> 3
44 -> 4
22 -> 2
55 -> 5
78 -> 8
88 -> 8
0
1
2
3
4
5
6
7
8
9
1
2
3
44
22
55
78
88
AfterAcademy

*

Thank You

*