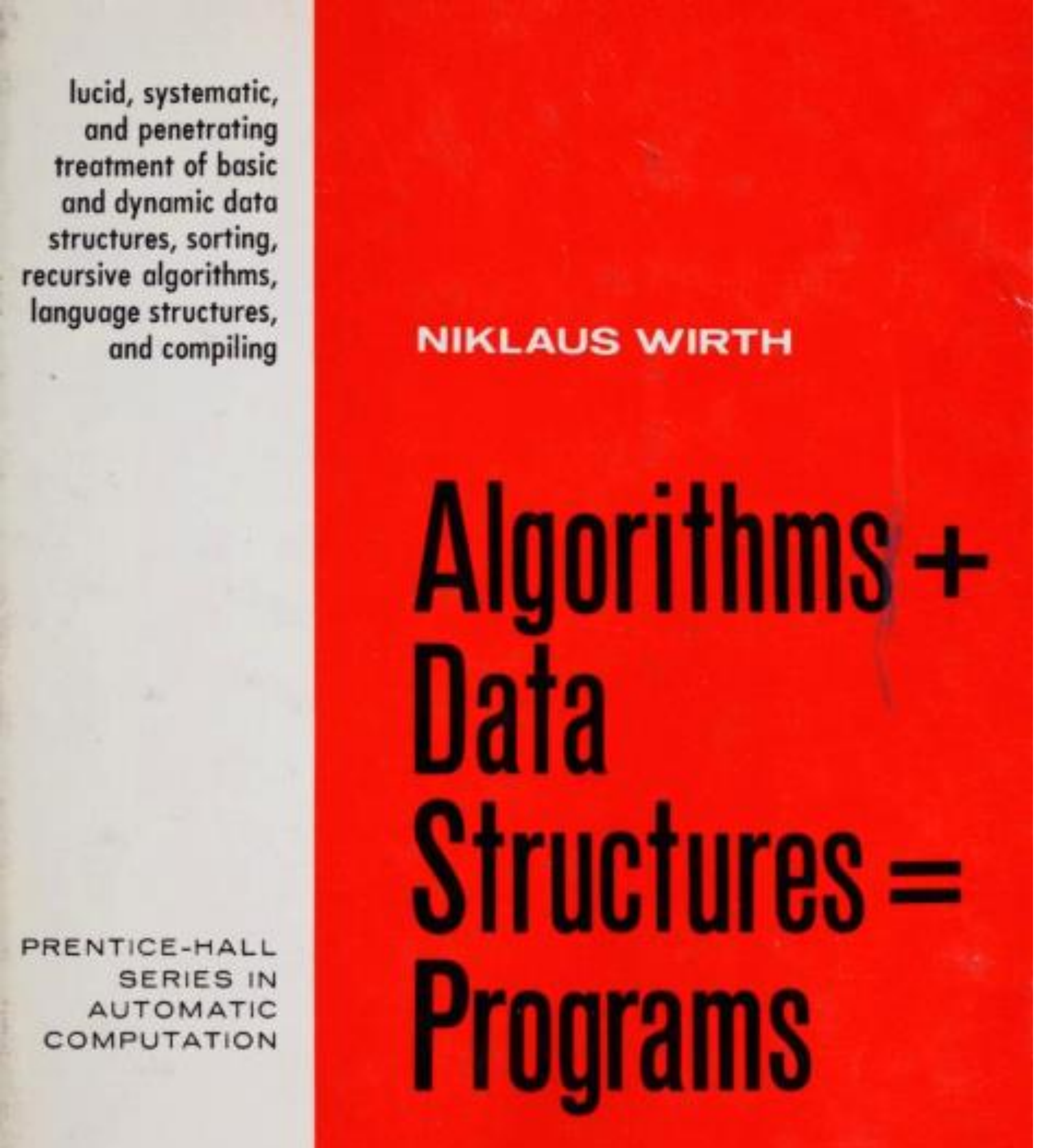
Fundamental Data Structures I

Dr Roberto Murcio

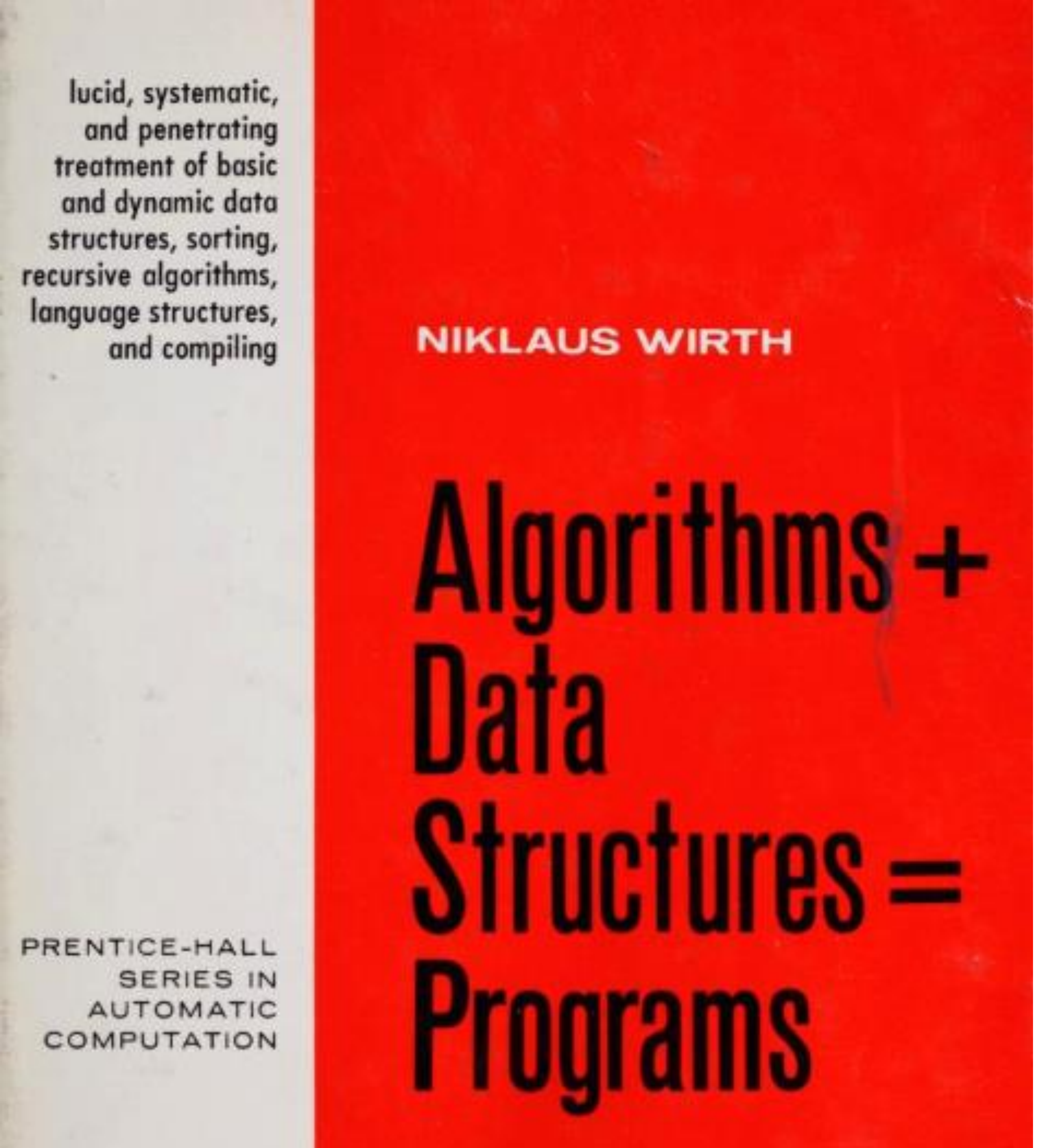


lucid, systematic, and penetrating treatment of basic and dynamic data structures, sorting, recursive algorithms, language structures, and compiling

PRENTICE-HALL
SERIES IN
AUTOMATIC
COMPUTATION

NIKLAUS WIRTH

Algorithms + Data Structures = Programs



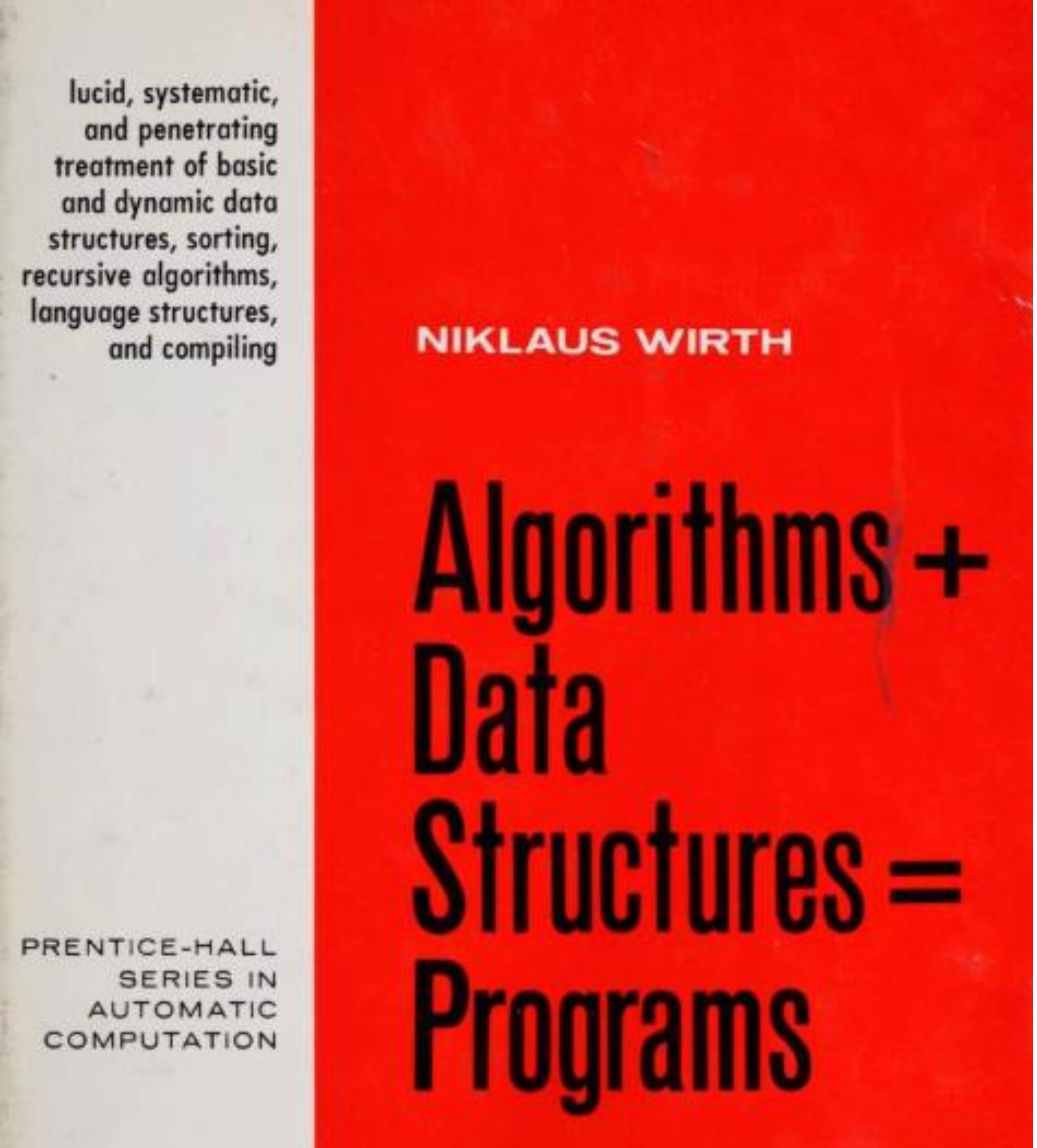
lucid, systematic,
and penetrating
treatment of basic
and dynamic data
structures, sorting,
recursive algorithms,
language structures,
and compiling

PRENTICE-HALL
SERIES IN
AUTOMATIC
COMPUTATION

NIKLAUS WIRTH

Algorithms +
Data
Structures =
Programs

Algorithm



lucid, systematic, and penetrating treatment of basic and dynamic data structures, sorting, recursive algorithms, language structures, and compiling

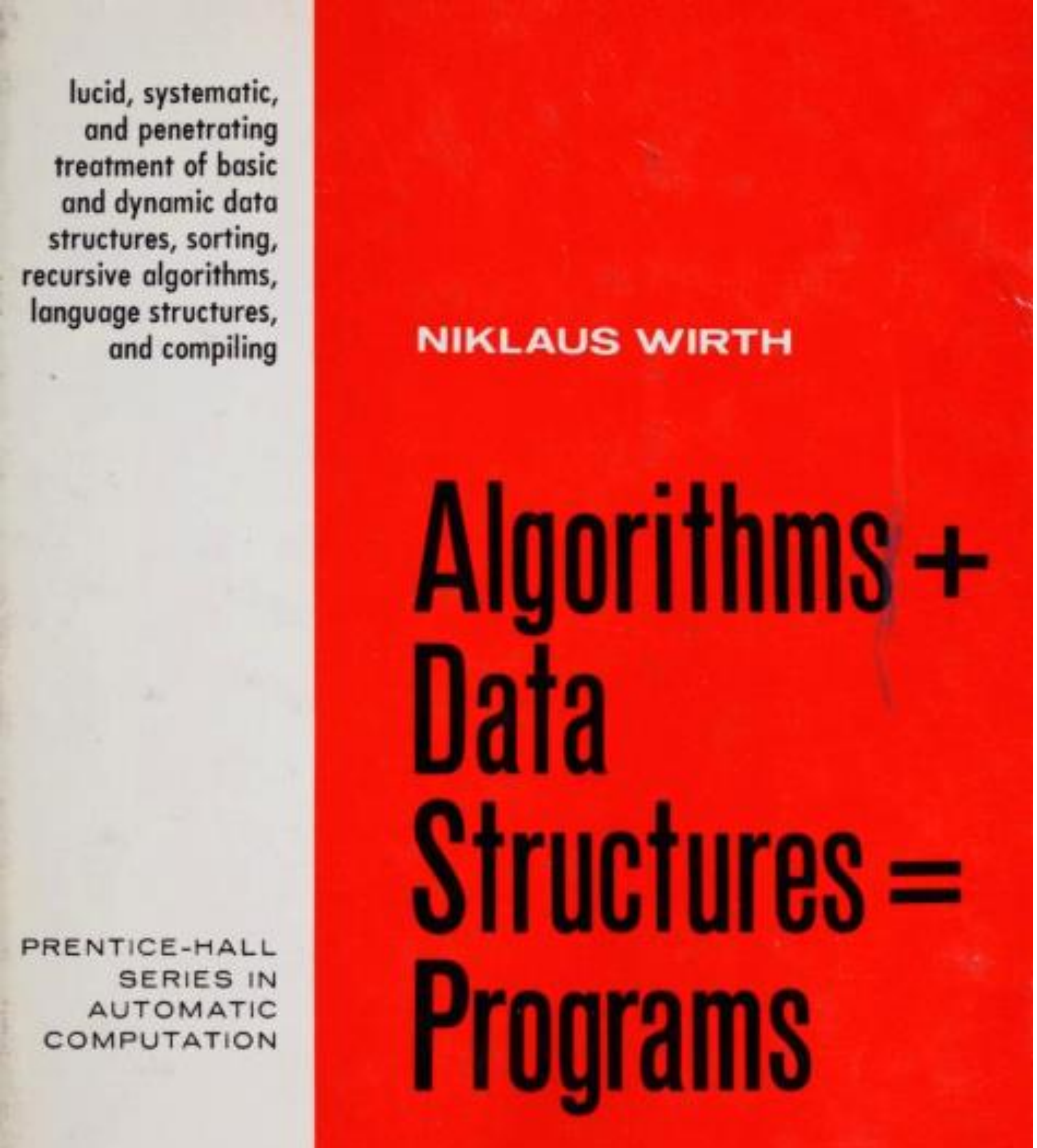
NIKLAUS WIRTH

Algorithms + Data Structures = Programs

PRENTICE-HALL SERIES IN AUTOMATIC COMPUTATION

Algorithm

Any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output.



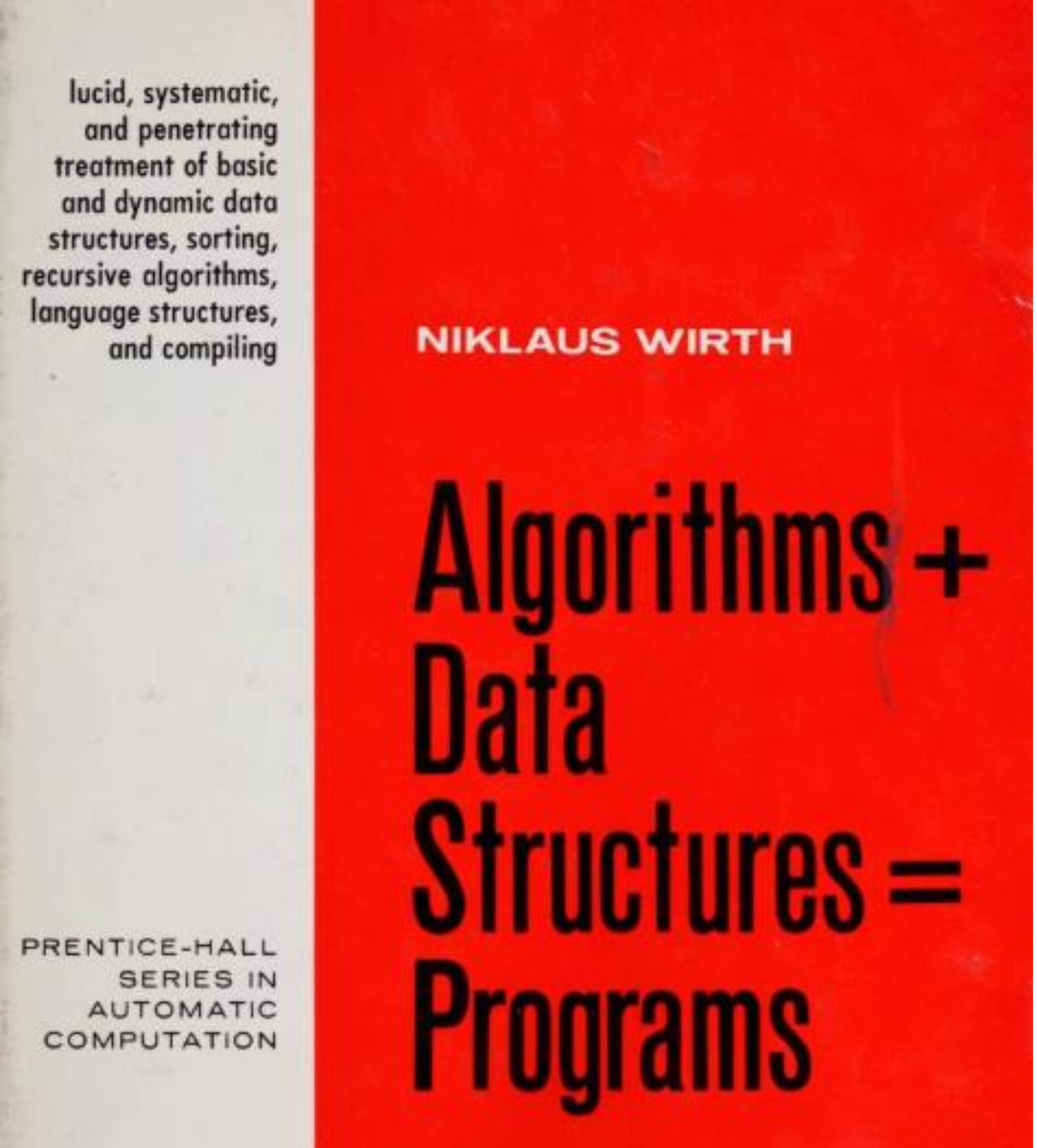
lucid, systematic, and penetrating treatment of basic and dynamic data structures, sorting, recursive algorithms, language structures, and compiling

NIKLAUS WIRTH

Algorithms + Data Structures = Programs

PRENTICE-HALL SERIES IN AUTOMATIC COMPUTATION

Algorithm
Data Structures (?)



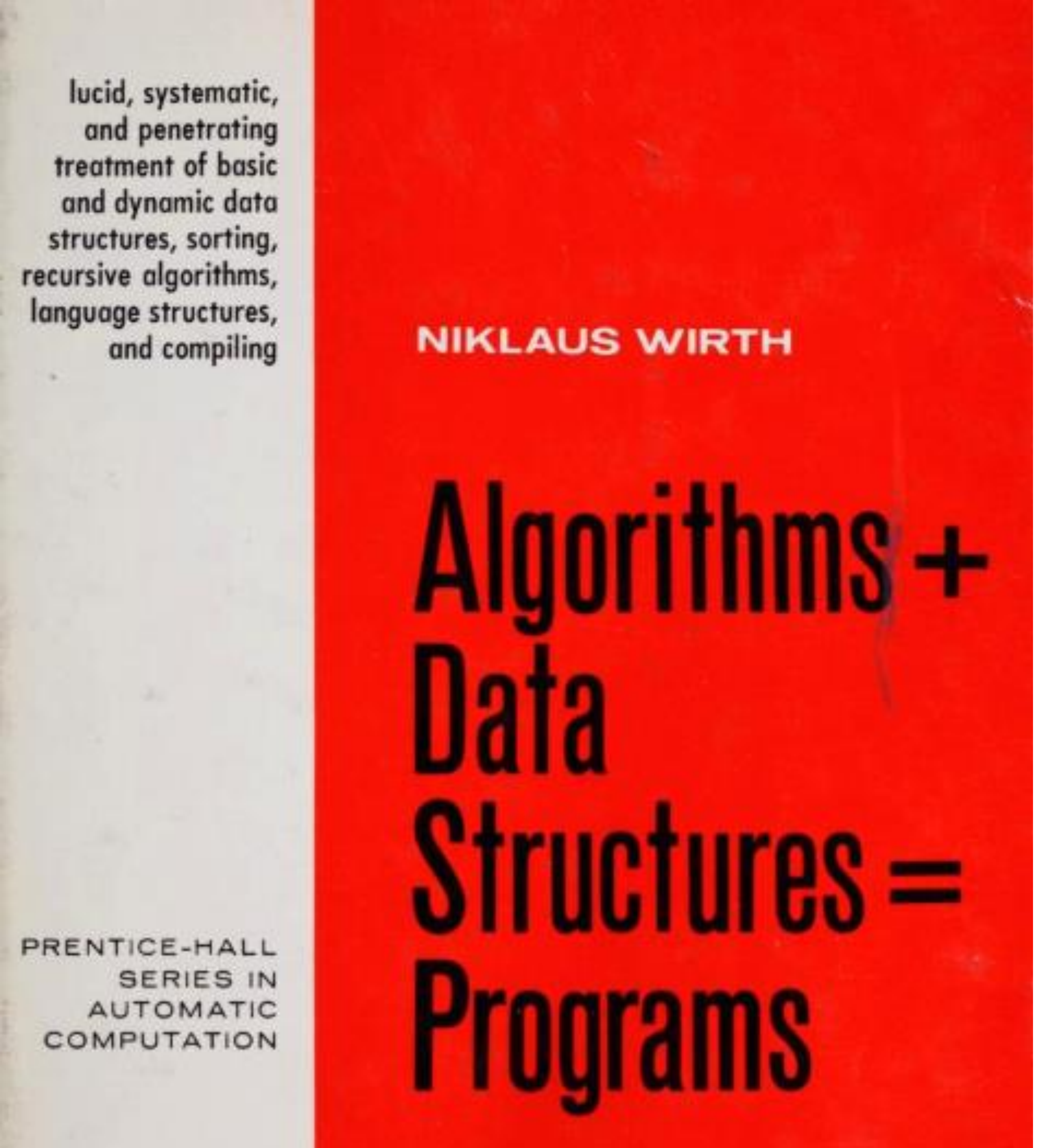
lucid, systematic, and penetrating treatment of basic and dynamic data structures, sorting, recursive algorithms, language structures, and compiling

NIKLAUS WIRTH

Algorithms + Data Structures = Programs

PRENTICE-HALL SERIES IN AUTOMATIC COMPUTATION

Algorithm
Data Structures
Programs



lucid, systematic,
and penetrating
treatment of basic
and dynamic data
structures, sorting,
recursive algorithms,
language structures,
and compiling

NIKLAUS WIRTH

Algorithms +
Data
Structures =
Programs

PRENTICE-HALL
SERIES IN
AUTOMATIC
COMPUTATION

Algorithm
Data Structures
Programs

A computer program is a collection of instructions that can be executed by a computer to perform a specific task to form a specific task.

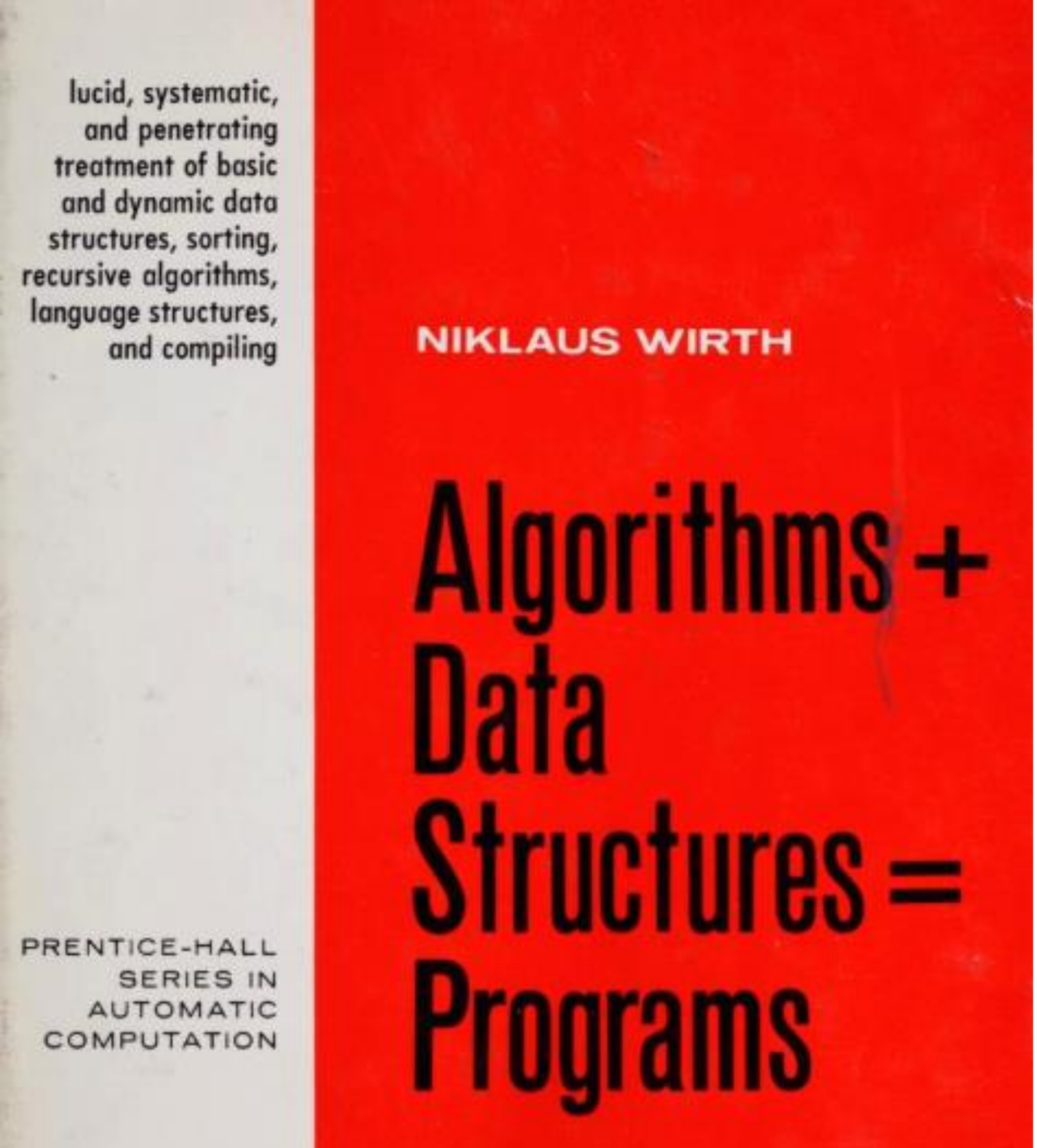
Data Structures

A data structure is a specialized format for organizing, processing, retrieving and storing data.

Data Structures

A data structure is a specialized format for organizing, processing, retrieving and storing data.

Data structure is the arrangement of data in memory



lucid, systematic, and penetrating treatment of basic and dynamic data structures, sorting, recursive algorithms, language structures, and compiling

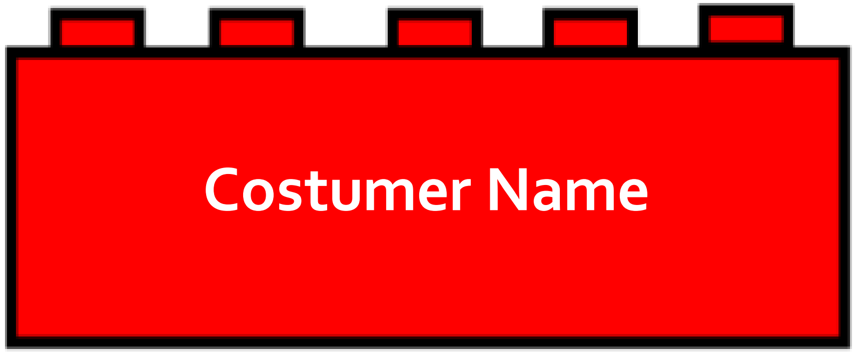
NIKLAUS WIRTH

Algorithms + Data Structures = Programs

PRENTICE-HALL
SERIES IN
AUTOMATIC
COMPUTATION

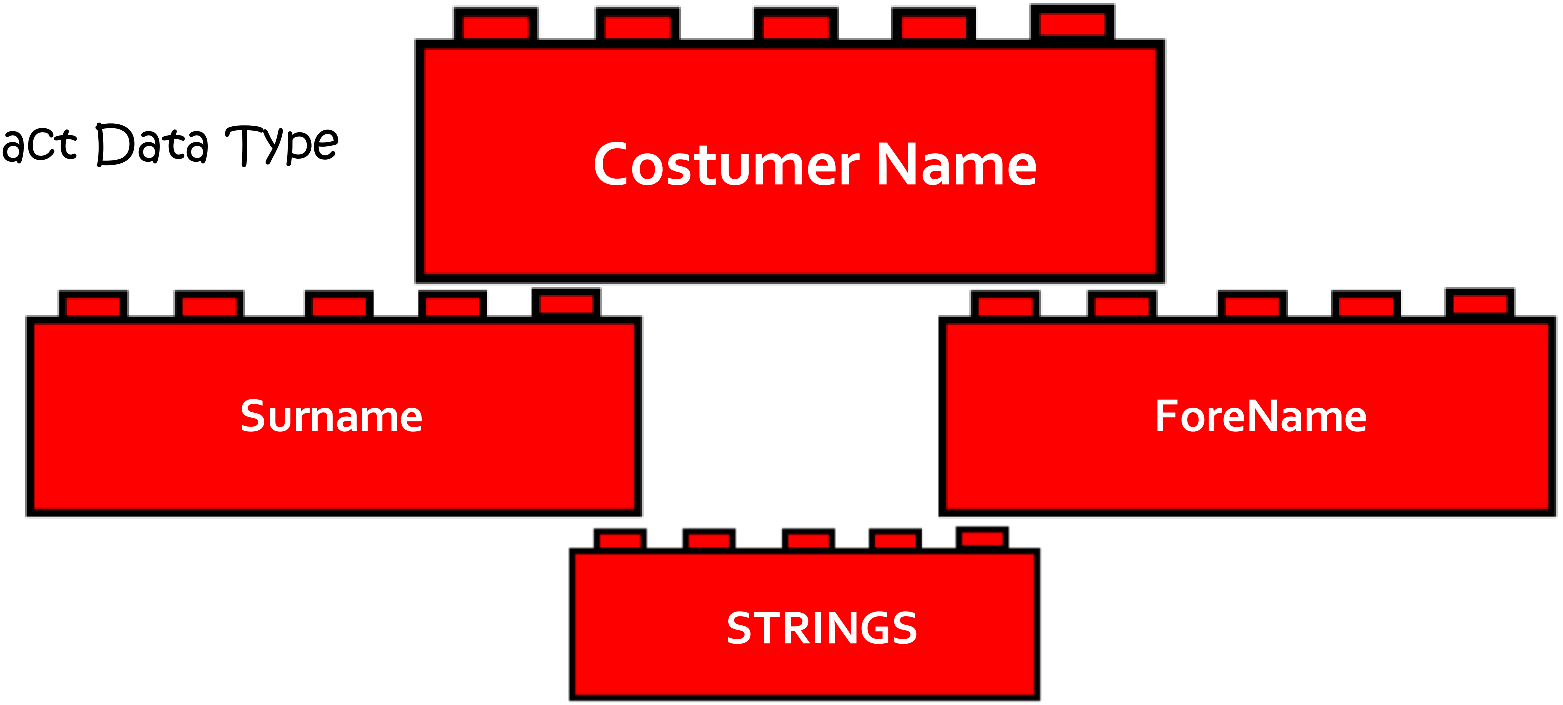
Algorithm
Data Structures
Programs

Non-trivial programs ALWAYS involve an algorithm supported by a data structure



Data structures are the building blocks for more sophisticated applications.

Abstract Data Type



Costumer Name

Surname

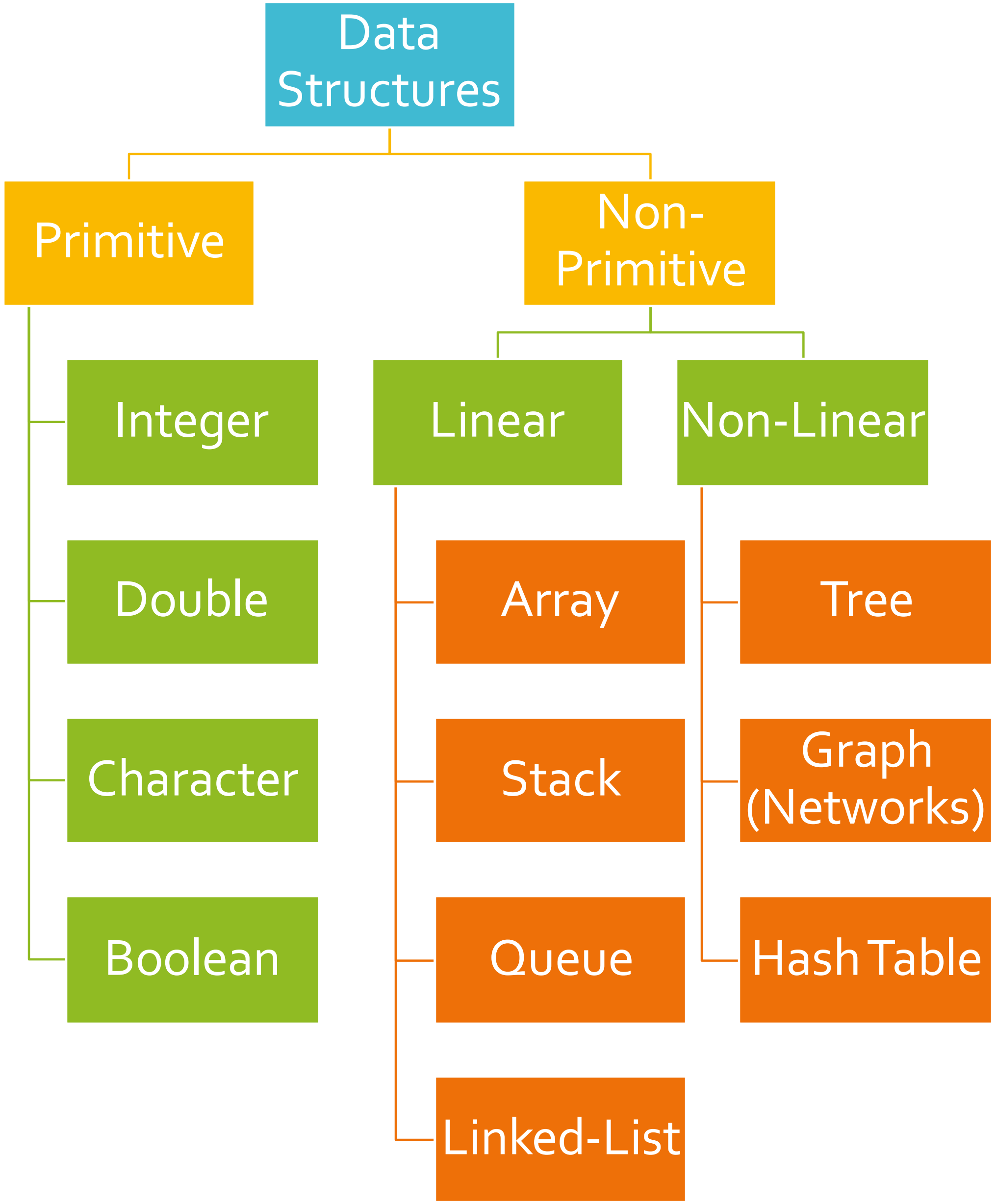
ForeName

STRINGS

LIST of characters

Data structures are the building blocks for more sophisticated applications.

Taxonomy of Data Structures



Five factors to consider when picking a data structure

WHAT KIND OF INFORMATION WILL BE STORED?

HOW WILL THAT INFORMATION BE USED?

WHERE SHOULD DATA PERSIST, OR BE KEPT, AFTER IT IS CREATED?

WHAT IS THE BEST WAY TO ORGANIZE THE DATA?

WHAT ASPECTS OF MEMORY AND STORAGE RESERVATION MANAGEMENT SHOULD BE CONSIDERED?

Software Development 1

Lab 5 – Data Manipulation in Python	Lesson 1: Lists
Lab 7 – Functions in Python	Lesson 2: Dictionaries
Lab 2 – Introduction to fundamental concepts of algorithms	Lessons 1 and 2

Abstract Data Types

Everything in a computer program came from an ADT

ADT are the theoretical concept behind algorithms and data structures.

ADT define the behaviour of a structure by a set of operations, without specify how to implement such operations.

Abstract DataType

Class of objects whose logical behavior is defined by a set of values and a set of operations

Dale, Nell; Walker, Henry M. (1996). *Abstract Data Types: Specifications, Implementations, and Applications*. Jones & Bartlett Learning

Primitive ADT

- Integers
  - Values in {…,-3,-2,-1,0,1,2,3,…}
  - Operations in [+, -, ×, ÷, ≤, ≥, …]
- These operations induced a behaviour, for example, commutativity for x and npozero division for ÷.
- This definition for the integer ADT is INDEPENDENT from the type of computer or programming language

ADT operations

Creators create new objects of the type.

Producers create new objects from old objects of the type.

Observers take objects of the abstract type and return objects of a different type.

Mutators change objects.

Integers in C : *int* is C primitive integer type

creators: the numeric literals 0, 1, 2, …

producers: arithmetic operators +, -, ×, ÷

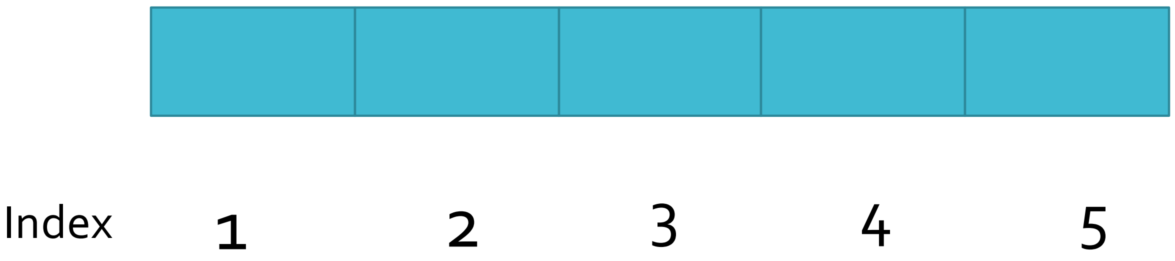
observers: comparison ==, !=, <, >

mutators: none

Archetypical Examples

LIST

get()
insert()
remove()
replace()

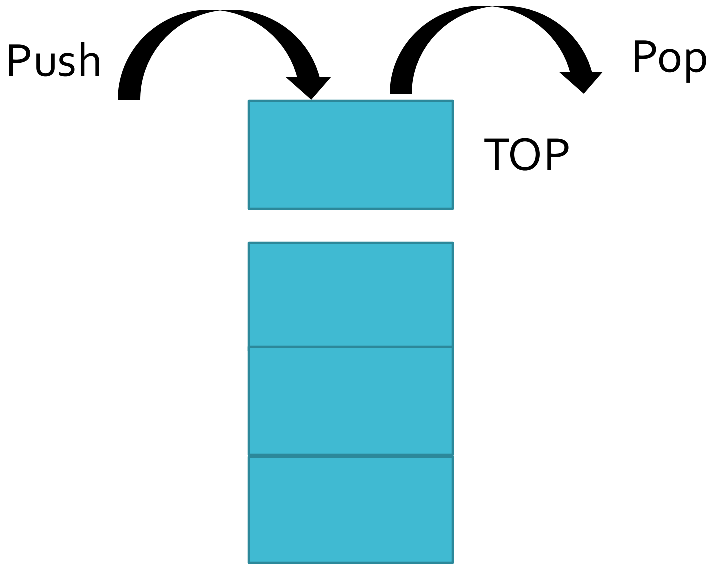


Index 1 2 3 4 5

size()
isEmpty()

A Sequence of objects in an ordered fashion

STACK



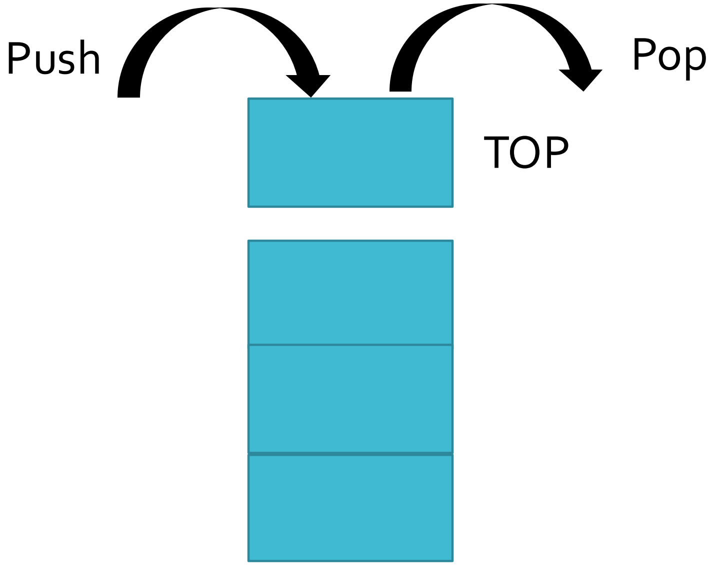
push()
pop()

size()
isEmpty()
top()

Sequential data structure which maintains the order of elements as they were inserted

Archetypical Examples

STACK



push()
pop()

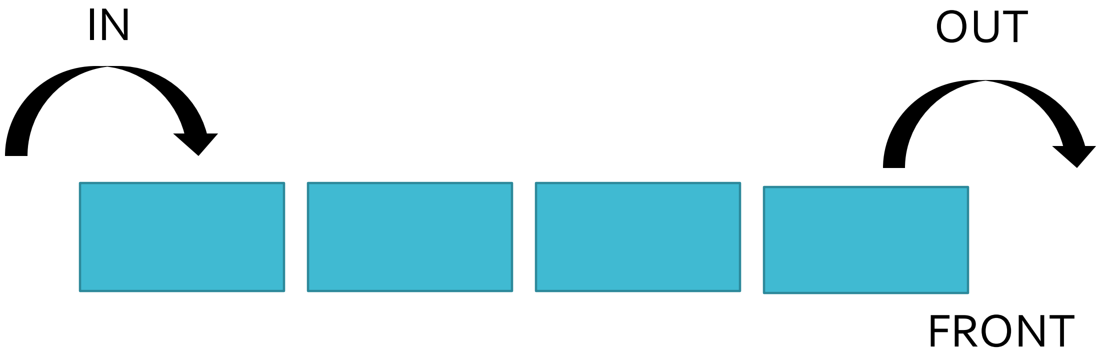
size()
isEmpty()
top()

LIFO

Sequential data structure whic tains
the order of elements as they ere erted

QUEUE

enqueue()
dequeue()

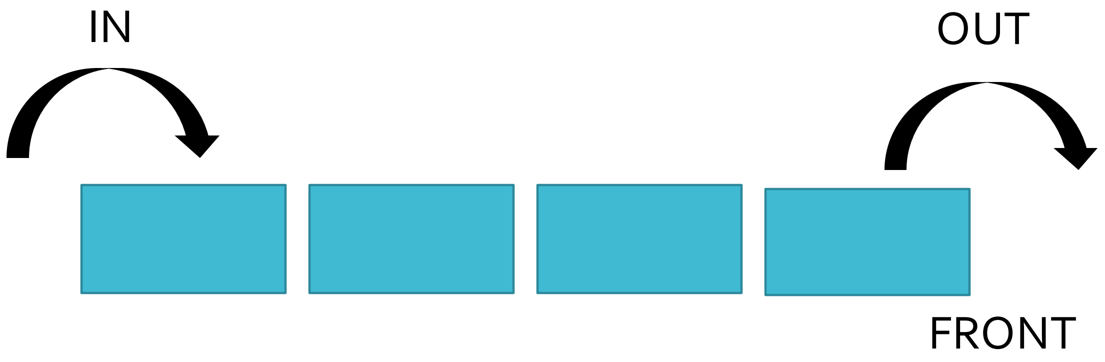


size()
isEmpty()
front()

Sequential data structure which maintains the order of elements as they were inserted

QUEUE

enqueue()
dequeue()



size()
isEmpty()
front()

Sequential data structure whic tains
the order of elements as they ere erted

FIFO

ADT
One definition, many implementations

- These definitions do not specify how these ADTs will be represented and how the operations will be carried out.
- List ADT can be implemented using arrays, or linked list
- Stack and Queue ADTs can be implemented using arrays or linked lists.

Summary

- Data structures + Algortims = Programs
- Theres are more than one type of data structure
- Abstract Data Types are one of the foundations of CS
- ADT define the behavioural aspect of a structure but no its implementation

THANKS!