Introduction to Data Structures and Algorithms

Dr. Sirasit Lochanachit

Outline

1. Resources
2. What is a “Data Structure”?
3. What is an “Algorithm”?
4. Prerequisites
5. Lesson Plan
6. Grading

Resources

1. Course Website:
   https://github.com/noswolf/DSA_BIT/tree/DSAP_24
2. Google Colaboratory
   - Interactive notebooks (.ipynb)
   - https://colab.research.google.com/
   - Login with your kmitl email address

What is a "Data Structure" ?

How do we store, organise, and retrieve data on a computer?

What is a "Data Structure" ?

- Way to **store** and **organise** data
- Enable efficient **access** and **modification** of data
- Designed for a specific algorithm
  - Strengths and limitations
  - Time and space complexity

Abstract Data Type

- A data type where only **behavior** is defined but not implementation.
- Examples: Array, List, Map, Queue, Set, and etc.

Common vs Abstract Data Type

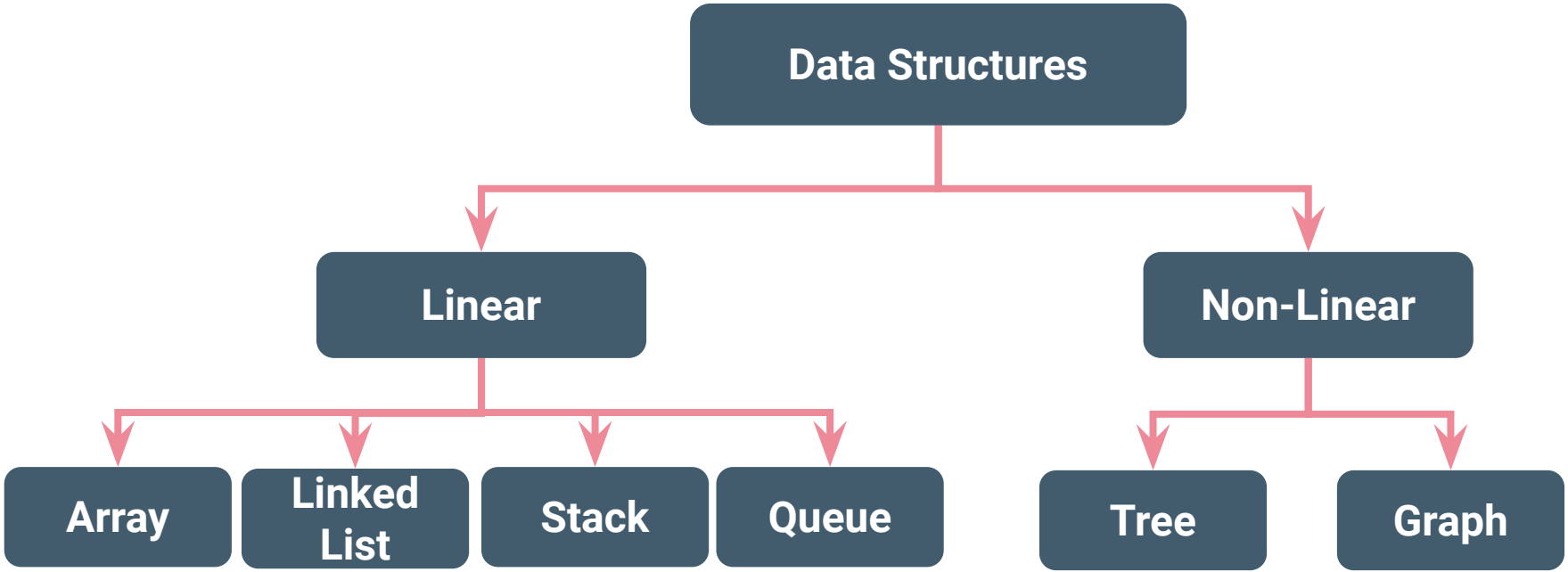
Common

- Integer
- Floating-point number
- Character
- String
- Boolean
- etc.

Abstract

- Array
- List
- Map
- Queue
- etc.

Type of Data Structure



Check out for a comprehensive list of data structures at
https://en.wikipedia.org/wiki/List_of_data_structures

What is an "Algorithm" ?

- Well-defined procedure or set of instructions to
  - transform input to output or
  - accomplish a task or
  - solve a computational problem

Why care about an "Algorithm" ?

How can we efficiently (in space/time) carry out some typical data processing operations?

How do we analyze and describe their performance?

Example: Sorting numbers

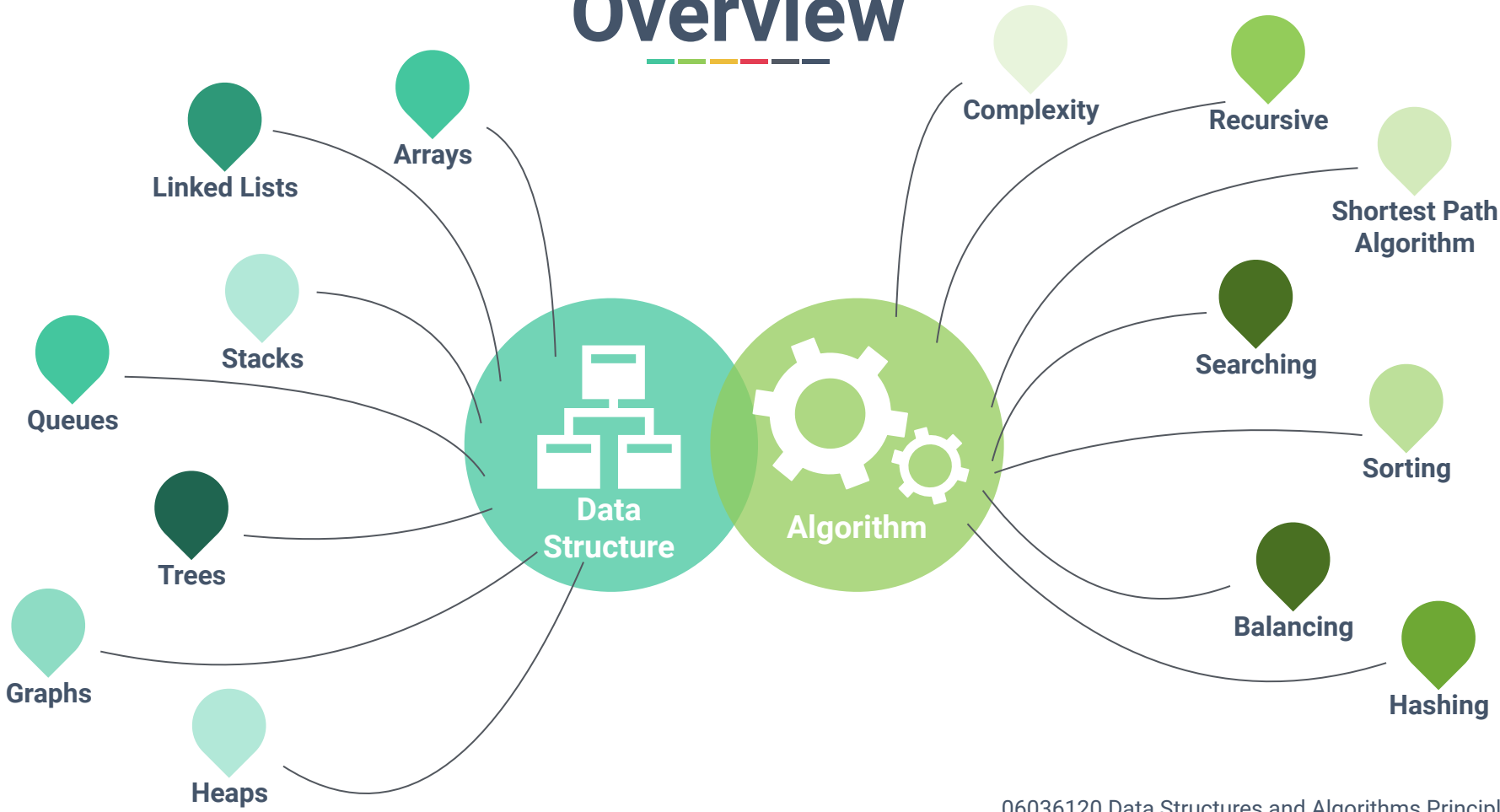
1. Input:

2. Sorting Algorithms
3. Output:

What kind of problems are solved by algorithms?

- Human Genome Project
  - identifying all genes of human beings
- Internet: Routing, searches, and security
  - **Shortest path**, search engines, encrypted communication
- E-commerce
  - Ads, recommendations, authentications
- Commercial enterprises
  - Resource allocation:
    - crew assignment on flights, package delivery route

Overview


Linked Lists
Arrays
Stacks
Queues
Trees
Graphs
Heaps
Data Structure
Algorithm
Complexity
Recursive
Shortest Path Algorithm
Searching
Sorting
Balancing
Hashing

Summary

- Data Structure
  - Way to store and organise data, allowing operations to be performed efficiently.
- Algorithm
  - Step-by-step procedure, which performs on data structure, to be followed to solve a problem/accomplish a task.

Prerequisites

- Fluent in Python Programming
- Comfortable with development processes
  - Writing a function
  - Debugging and testing a code

Lesson Plan (till Midterm)

*Public holidays are highlighted in red

Week	Topics	Individual Assignments
01/07/2024	Introduction	#1 Python Crash Course
08/07/2024	Algorithm Analysis	#2
15/07/2024	Arrays	#3
22/07/2024	Stacks [VDO]	#4
29/07/2024	Queues [VDO]	
05/08/2024	Binary Search	#5
12/08/2024	Linked Lists [VDO]	#6
19/08/2024	Trees	-

Lesson Plan (after Midterm)

*Public holidays are highlighted in red

Week	Topics	Individual Assignments
02/09/2024	Binary Search Trees	#7
09/09/2024	Hashing	#8
16/09/2024	Sorting	#9
23/09/2024	Recursion and Sorting	
30/09/2024	Graphs	-
07/10/2024	Graphs (Cont.)	#10
14/10/2024	-	-
22/10/2024	Final Exam	

Grading

Attendance	5%
Individual Assignment	25%
Midterm Exam	35%
Final Exam	35%

Attendance Score

Out of 10 times, If you attend onsite

- 10 times = 5%
- 9 times = 4%
- 8 times = 3%
- 7 times = Unable to take a final exam

Leave notice must be submitted at least 48 hrs before the lecture.

Late Policy

Individual Assignments: Late submission is not allowed

- Hard deadline: normally 1 or 2 week(s) after posted

Reading List

Essential

Goodrich, M.T., Tamassia, R. and Goldwasser, M.H., 2013. ***Data structures and algorithms in Python***. John Wiley & Sons Ltd.

Recommended

Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C., 2022. ***Introduction to algorithms***. MIT press.

Miller, B.N. and Ranum, D.L., 2011. ***Problem solving with algorithms and data structures using python***, 2nd ed. Franklin, Beedle & Associates Inc.