جامعة بيرزيت
BIRZEIT UNIVERSITY

Introduction

Dr. Abdallah Karakra

Computer Science Department

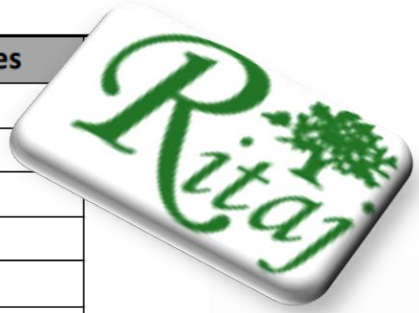
COMP242

Tuesday, February 27, 2024

Course Outline

Topics Covered in this Course:

	Description	# of Lectures
1	Recursion	2
2	Algorithm Analysis	4-5
3	Lists, Linked Lists, Double Linked Lists, examples	2-3
4	Cursor Implementation of Linked Lists, Stacks	2
5	Implementation of Stacks, Examples, Queues	3-4
	Project 1 + discussion	
6	Trees, Implementation of Trees, Binary Trees	2
7	Expression Trees, Binary Search Trees	2
	Midterm exam (30%) & Project 2 + discussion	
8	AVL Trees, Single and Double Rotation, Tree Traversals	2-3
9	Splaying, B_Trees	1
10	Hashing	2-3
	Project 3 + discussion	
11	Priority Queues (Heaps)	3
12	Sorting, Design Techniques, Merge Sort, Analysis of Merge Sort	2
13	Quick Sort , Analysis of Quick Sort, Linear Sorting Algorithm, Shell Sort, External Sort	3
	Final Exam (35%) & Project 4 + discussion	



Course Outline

References:

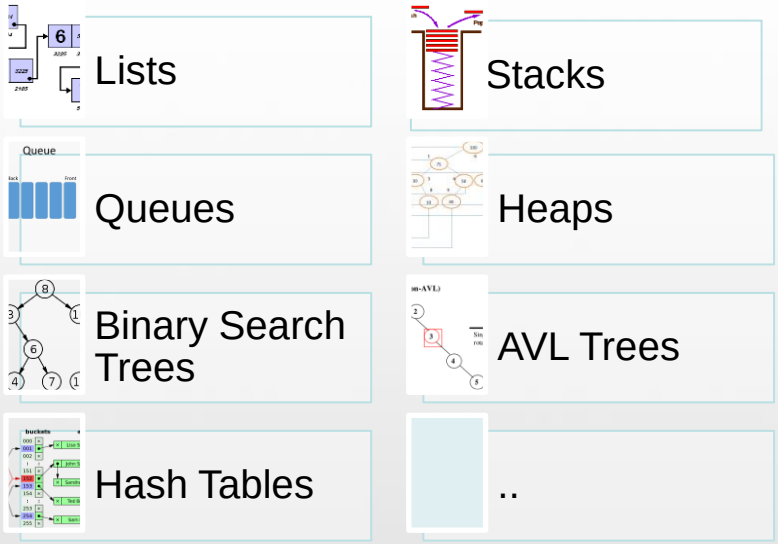
- **Data Structures and Algorithm Analysis in Java 3rd edition. Mark AllenWeiss, Pearson, 2011**
- **Laboratory Work Book** (COMP242)

Grading Criteria: (Tentative: this might change according to the teaching situation)

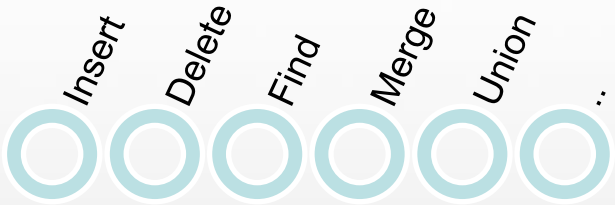
• Lab work (Quizzes and Participation)	15%
• Midterm exam	30%
• Projects	20%
• Final exam	35%

What will you learn in this course?

Clever ways to organize information in order to enable efficient computation.



Data Structures



Algorithms

Meaning: what does the operation do/return?
Performance: how efficient is the operation?

Terminology

- **Algorithm** :A high level, language-independent description of a step-by-step process
- **Abstract Data Type (ADT)** : Mathematical description of a “thing” with set of operations
- **Data structure** : A specific family of algorithms for implementing an ADT
- **Implementation of a data structure** : A specific implementation in a specific language on a specific machine (both matter!)

Data Structures

- How do we organize information so that we can find, update, add, and delete portions of it efficiently?

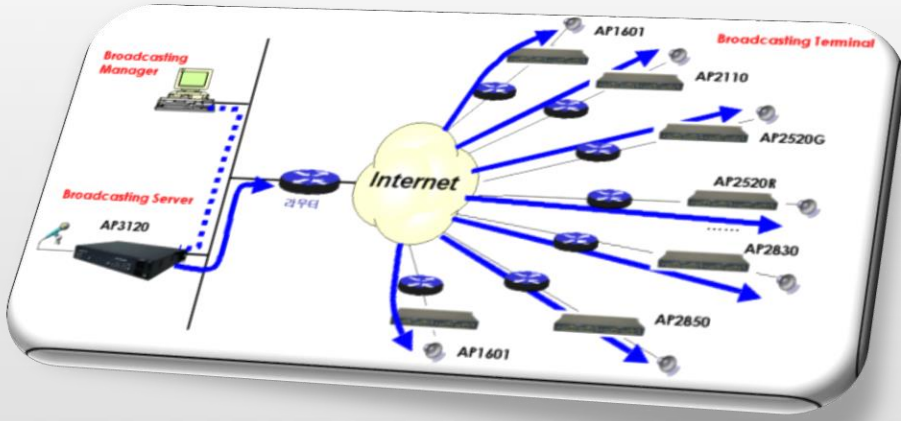
Running time

Space used

Have a look to the following Data Structure Example Applications

- How does Google quickly find web pages that contain a search term?

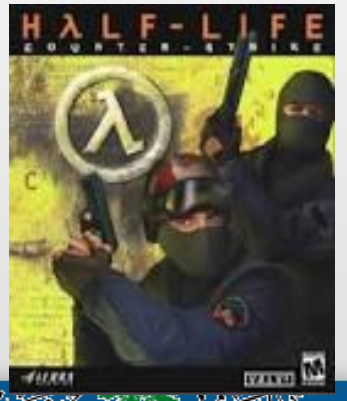
- What's the fastest way to broadcast a message to a network of computers?



Have a look to the following Data Structure Example Applications

- How does your operating system track which memory (disk or RAM) is free?

- In the game counter strike, how can the computer determine which parts of the scene are visible



- …

In other words, What is a Data Structure?

- How to store a collection of objects in memory?
- What operations we can perform on that data?
- What is the algorithms for those operations?
- How time and space efficient those algorithms are?

Example: Palestine's map

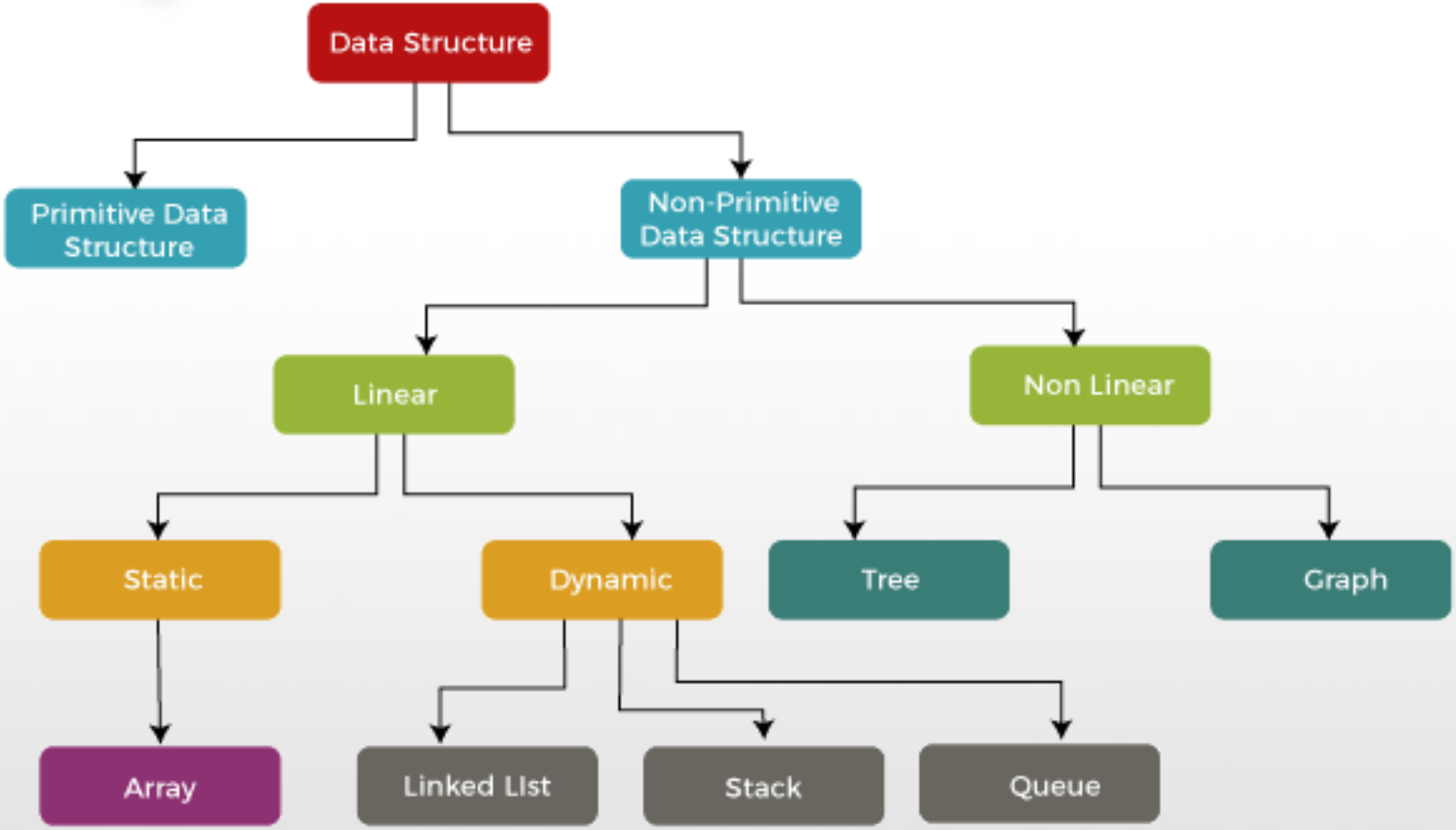
You want to store data about cities (location, elevation, population)...

What kind of operations should your data structure(s) support?

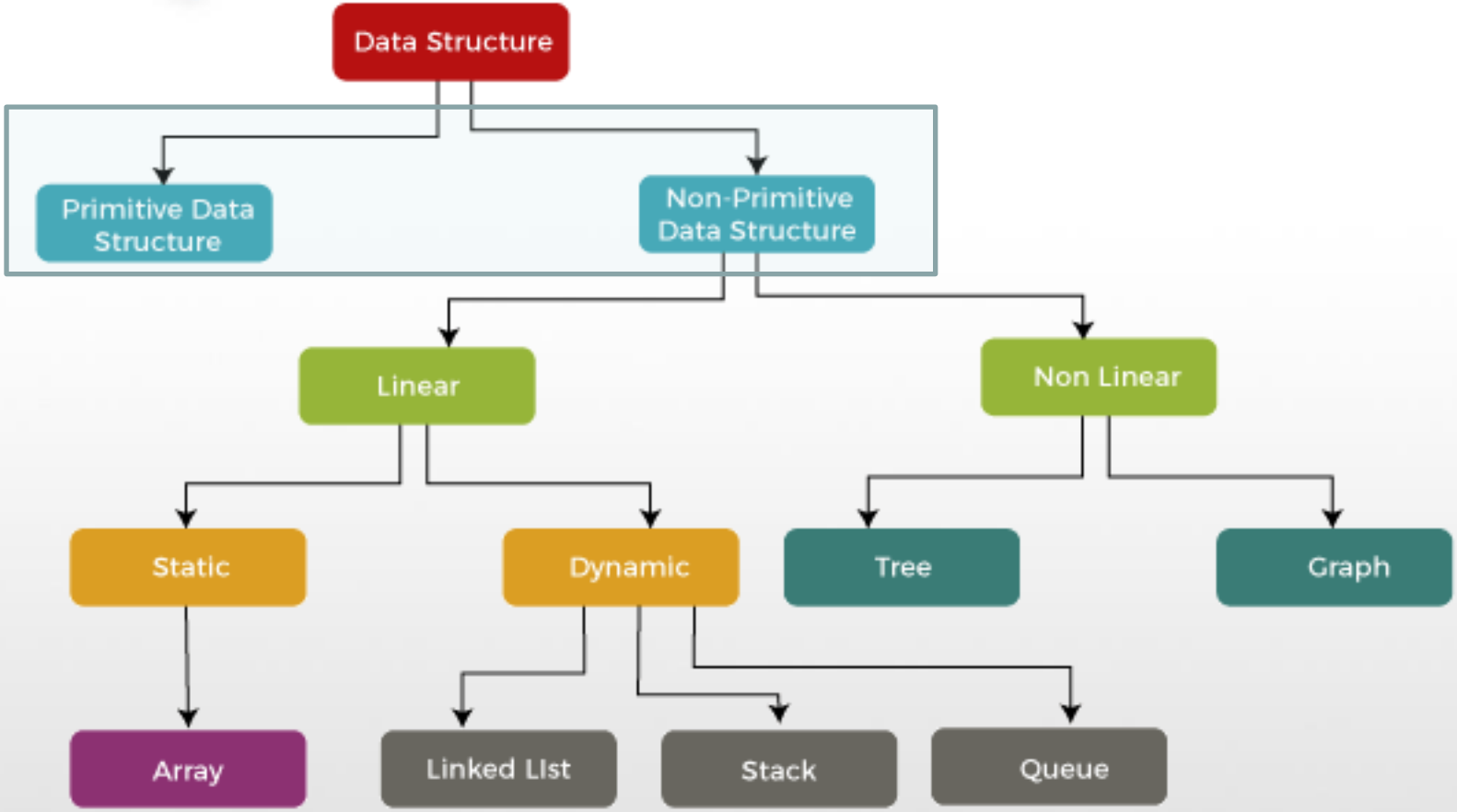
Example: Palestine's map

- Finding addresses on map?
  - Lookup city by name...
- Mobile iPhone user?
  - Find nearest point to me...
- Car GPS system?
  - Calculate shortest-path between cities...
  - Show cities within a given window...
- Political revolution?
  - Insert, delete, rename cities

Classifications of Data Structures



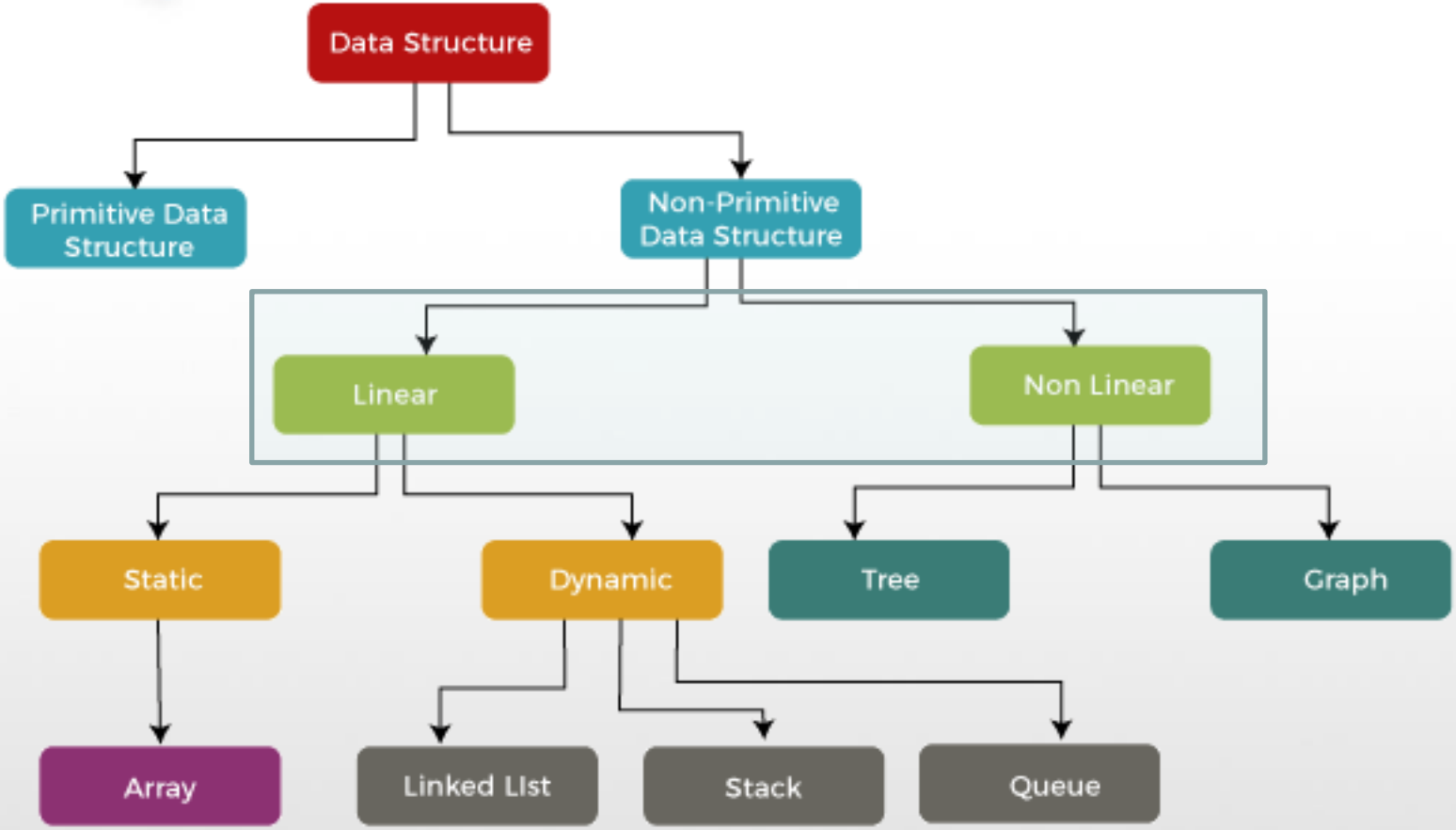
Classifications of Data Structures



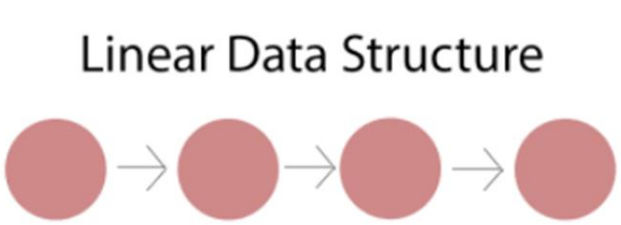
Classifications of Data Structures

- **Primitive data structure** : is a fundamental type of data structure that stores the data of only one type (**Integer,Float,etc)**
- **Non primitive data structures** : is a type of data structure which is a user-defined that stores the data of different types in a single entity

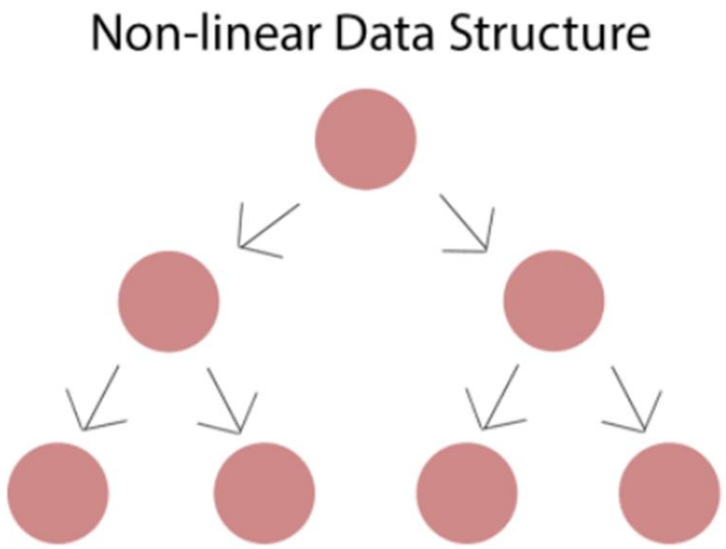
Classifications of Data Structures



Non primitive data structures



examples:
- arrays
- stacks
- queues
- linked lists

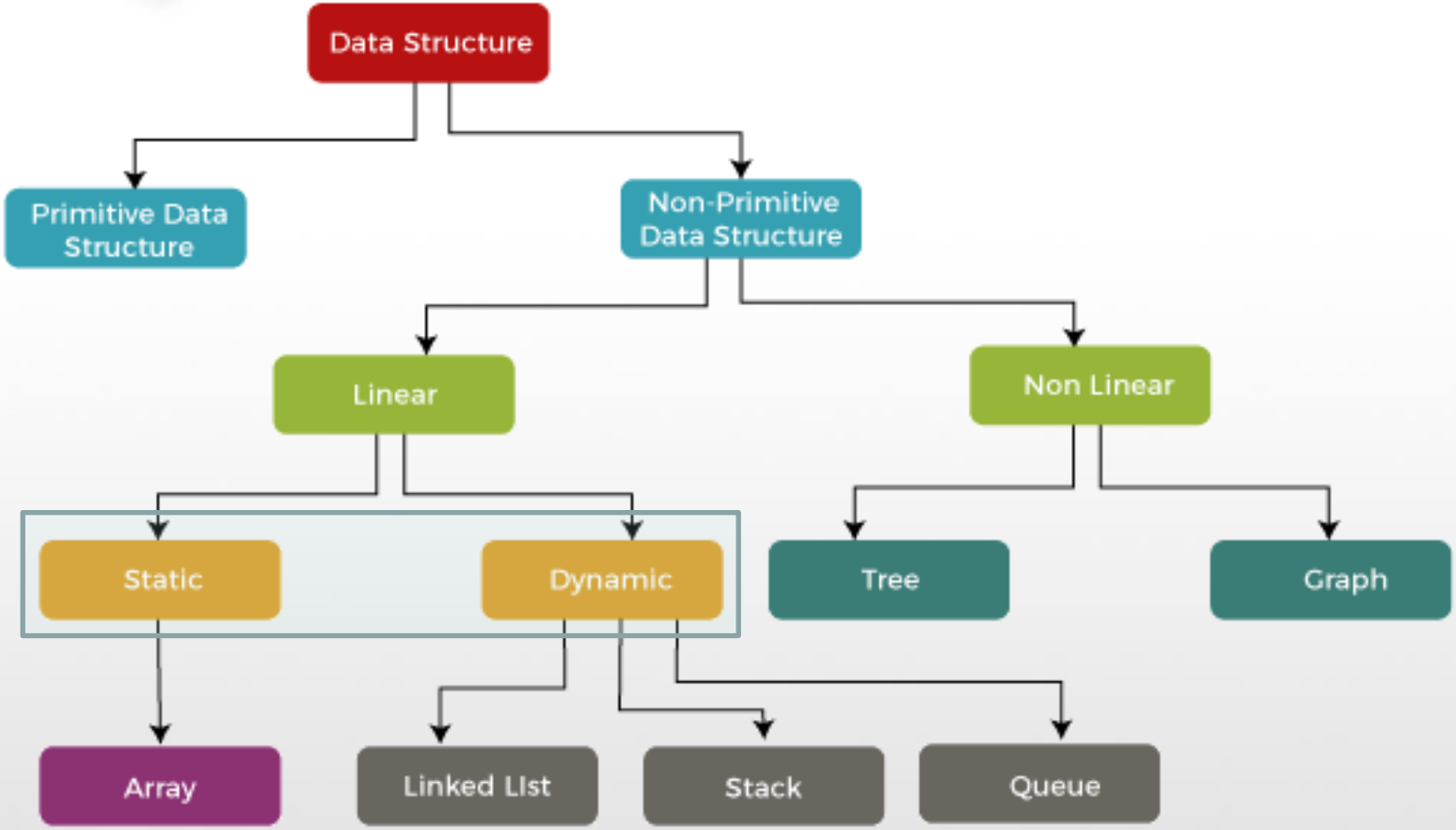


examples:
- trees
- graphs

Non primitive data structures

Factor	Linear Data Structures	Non-linear Data Structures
Arrangement of Data Element	data elements are sequentially connected	data elements are hierarchically connected
Travers of Data Element	each data element is traversable with a single run.	data elements are present at many levels
Examples	Array, Stack, Queue, Link List, etc	Graph, Tree, Map, etc
Presence on which Level?	all data elements are present at the same and single level.	data elements are present at multiple levels.
uses memory	not very memory friendly and are not utilizing memory efficiently.	very efficiently
Time Complexity	increases with an increase in size.	remain with an increase in size.
Complexity during Implementation	easier to understand and implement.	difficult to understand and implement

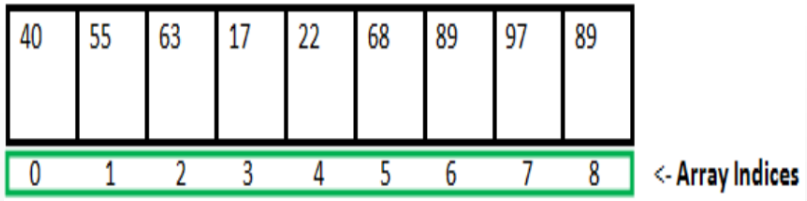
Classifications of Data Structures



Linear Data Structures

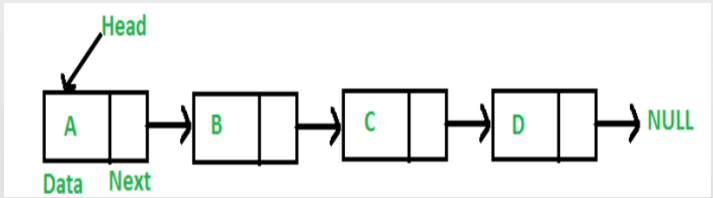
Static Data structure:

The size of the structure is fixed. The content of the data structure can be modified but without changing the memory space allocated to it.



Dynamic Data Structure:

The size of the structure in not fixed and can be modified during the operations performed on it. Dynamic data structures are designed to facilitate change of data structures in the run time.



Extra Exercises

Question?

“Success is the sum of small efforts, repeated day in and day out.”
Robert Collier

Reference: 1. Dave Mount's Lecture Notes
2. Carl Kingsford Lecture Notes
3. Dan Suciu Lecture Notes