CS202 – Data Structures

LECTURE-01

Overview of Data Structures

Motivation, Organizing Data for Applications

Dr. Maryam Abdul Ghafoor
Assistant Professor
Department of Computer Science, SBASSE

Register with “Poll Everywhere”

- Please register yourself using your LUMS email address: https://pollev.com/blesseddawn004/register

- To participate in activities, please open the following link on your browser and keep it open during class:
  - https://pollev.com/blesseddawn004/

Instructor: Dr. Maryam Abdul Ghafoor

- Assistant Professor of Computer Sciences
  - Ph.D., LUMS
  - Visiting Researcher at University College London, UK
- Courses
  - Data Structures, Databases, Software Engineering, and Coding for Careers at LUMS
- Research
  - I work with code optimization, malware detection, automated testing, bug similarities
  - Mental health challenges, AI in foundational computer science eduction

Momina Khan (Co-instructor CS 202)

A LUMS alum with a decade long teaching career

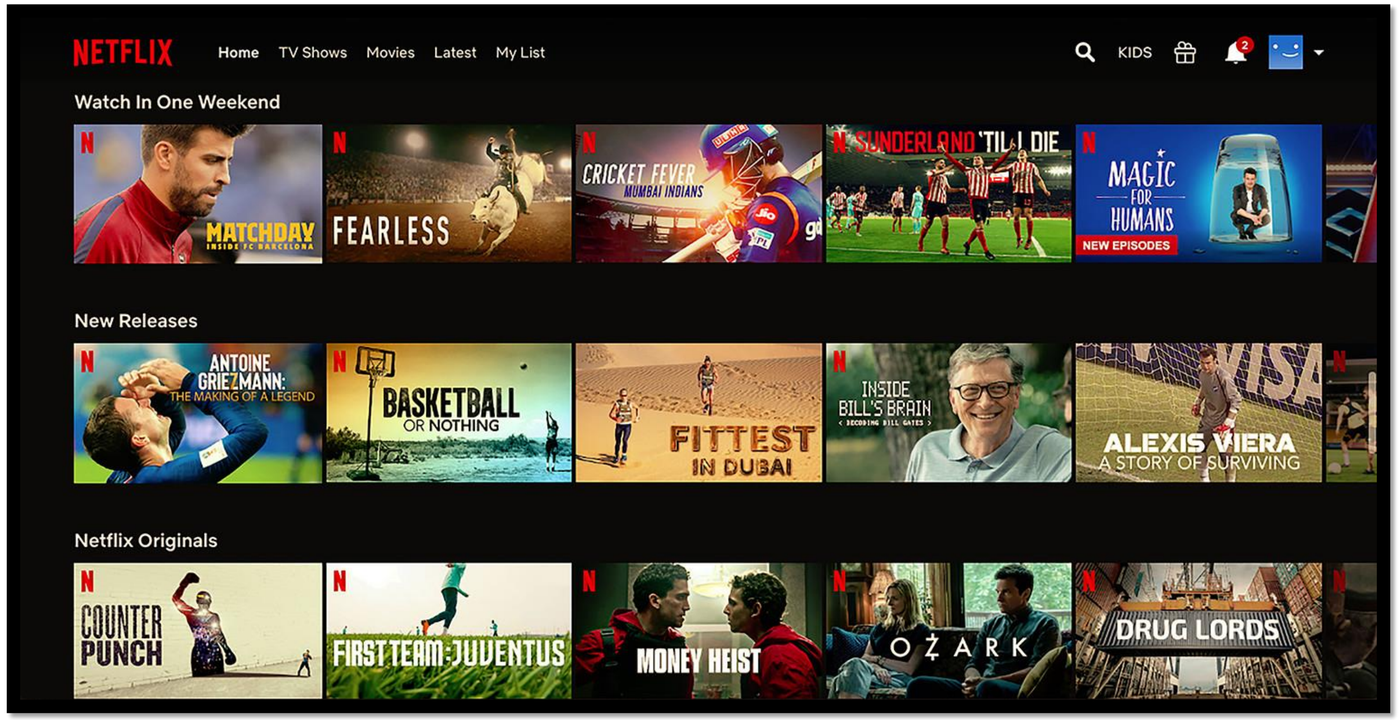
A keen and proficient swimmer

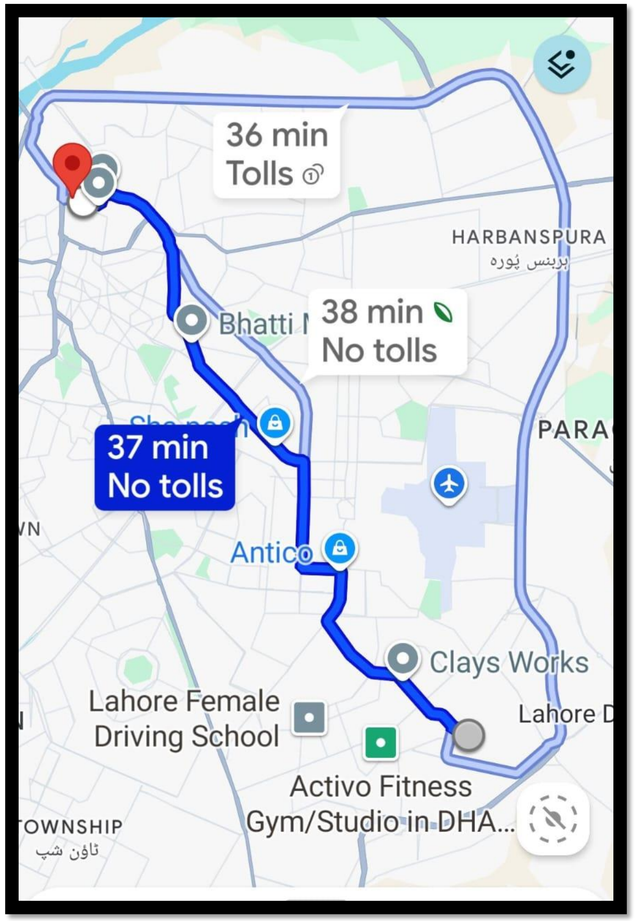
Data Structures is one of my favourite courses to teach ☺

Agenda

- Course Context & Motivation
- What are Data Structures?
- Course Administrivia
- Data Structures in the Era of Large Language Models
- Importance of Learning Data Structures

Under the Hood





Data to Decisions

- Many applications

- Wide variety of sources
  - Mobile apps, Website, IoT, wearable devices

Magic Behind the Apps

Behavior Data → Personalized Suggestions

Prompts → Responses

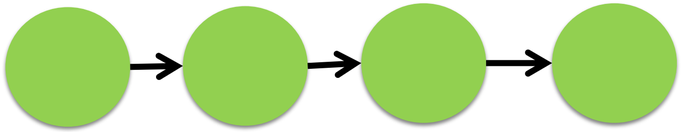
Health Data → Insights and Recommendations

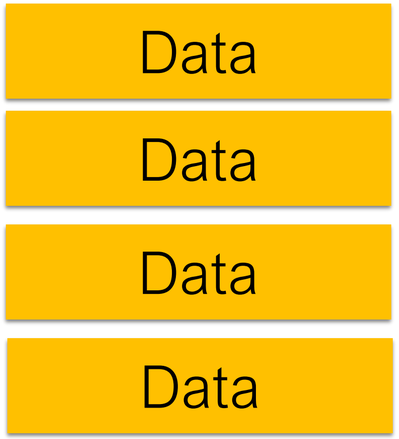
Personal and Social Info → Targeted Advertising

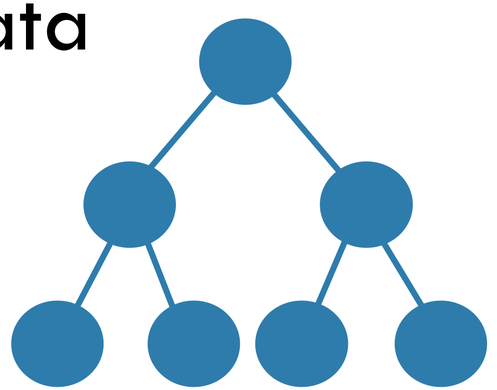
Data Structures

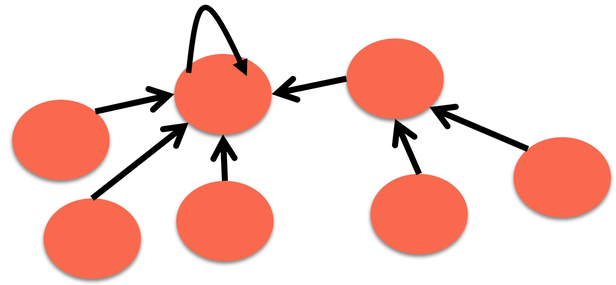
The science of "structures" for storing data

Data Collection









Data Organization

How should we structure our data so that it can be processed and stored efficiently?

What is a Data Structure?

- A systematic way of organizing data in a computer system

"There should be a method to the madness" :-)
Well-defined rules!

Structuring

Desktop, Smartphone, Cloud, Internet

Key Design Considerations

- Time
  - When data is stored in a structure, how long does the algorithm take to complete a task? (e.g., insert, delete, search)

- Space
  - How much space is taken by the structure?

- Other factors
  - (e.g., energy, network usage, user experience)

Why Time Efficiency?

Face recognition	370ms - 620ms
Speech recognition	300ms – 450ms
VR head tracking	< 16ms

Google

Traffic reduced by 20% due to 500ms increase in latency

[M. Mayer at Web 2.0]

amazon.com

Every 100ms latency costs 1% in business revenue

[Speed matters, G. Linden]

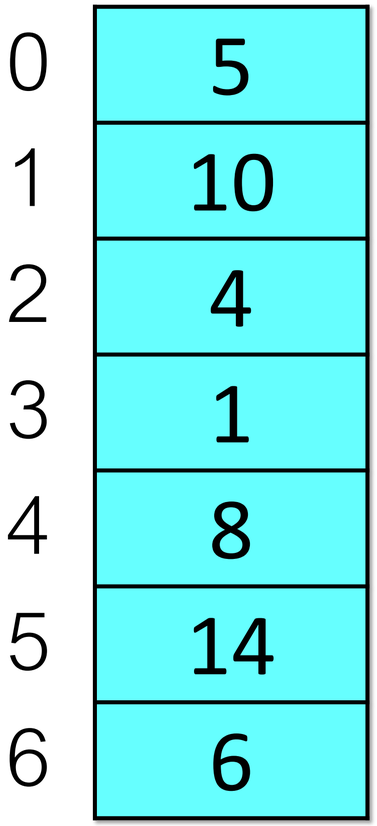
YAHOO!

An extra 400ms reduced traffic by 9%

[YSlow 2.0, S. Stefanov]

Why Structure Matters?

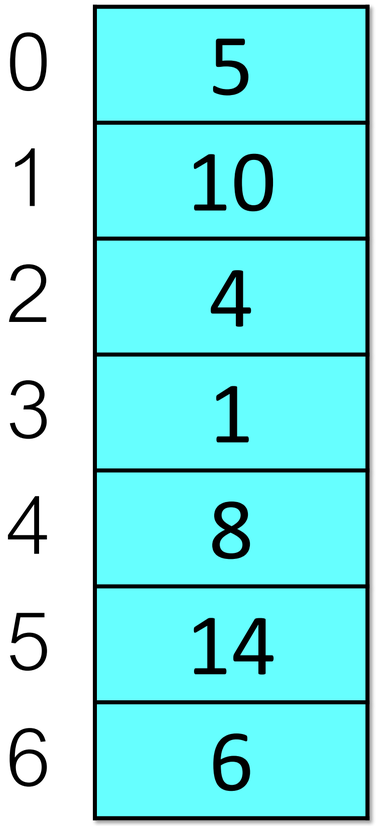
Activity



Task	Steps
`min()`	?
`find(x)`	?

Why Structure Matters?

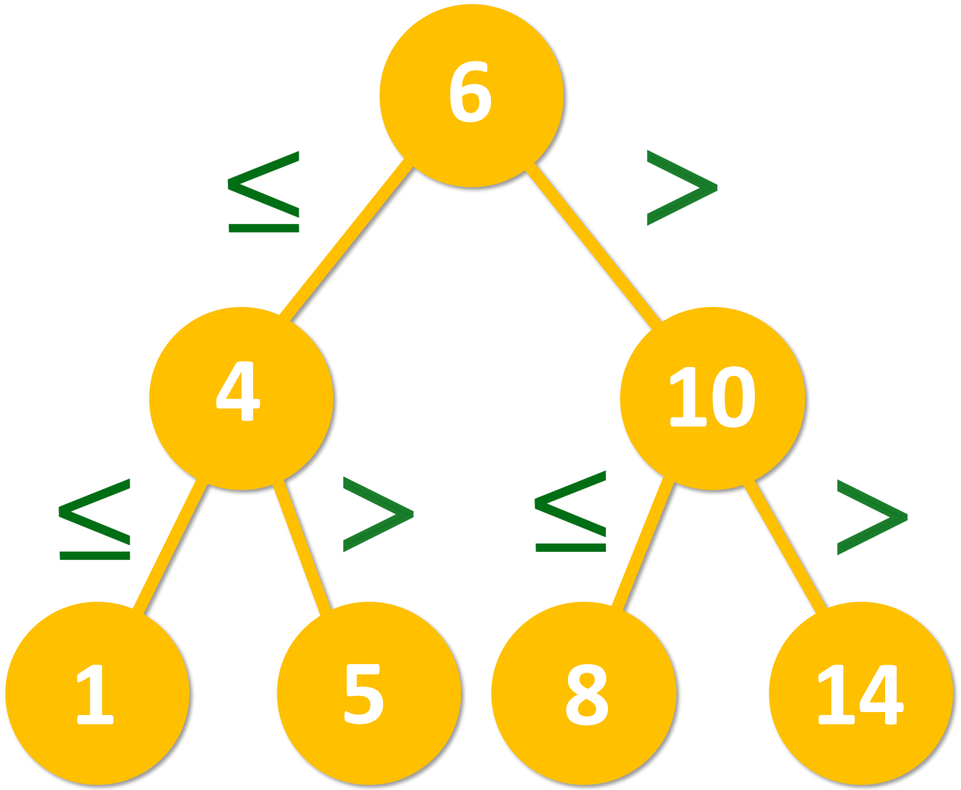
Activity



Task	Steps
`min()`	7
`find(x)`	4

Why Structure Matters?

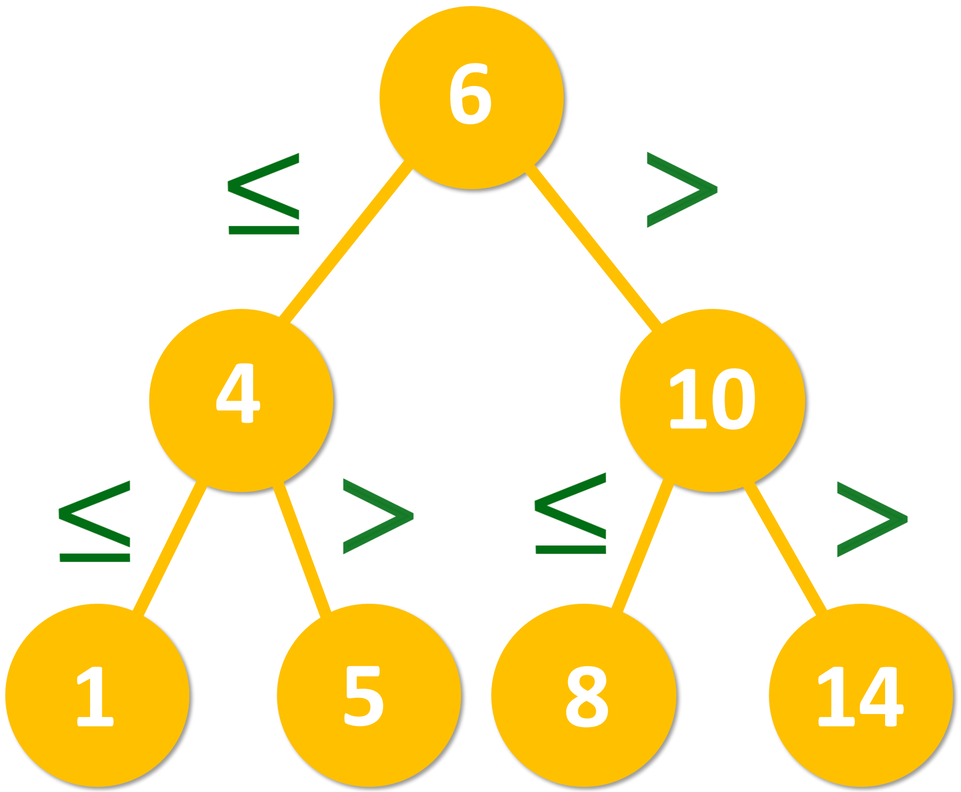
Activity



Task	Steps
`min()`	?
`find(x)`	?

Why Structure Matters?

Activity

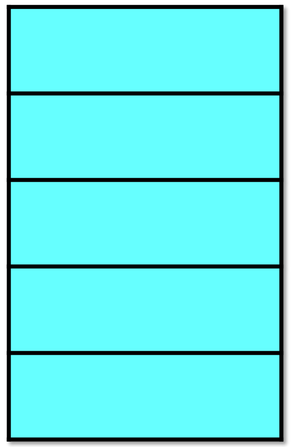


Task	Steps
`min()`	3
`find(x)`	[1+2(2)+4(3)]/7=2.4

Tree structure improves the number of steps needed to complete find(x) and min( ) operations by 1.7x and 2.3x, respectively

Generalizing….

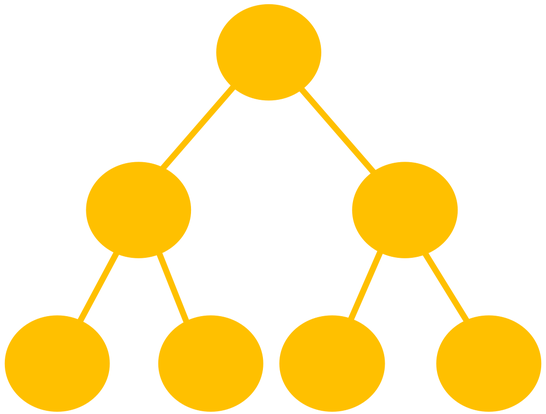
Linear Structure



~n

$\sim 10^9$

Tree Structure

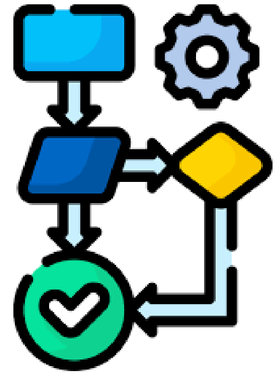


~ log(n)

30 Steps

In this course….

- How do we organize data?
  - What are the most important task for my application?
  - What are the most reoccurring tasks?



- How do we analyze our code?
  - Analyze and compare algorithms
  - What is the best case?, etc.
  - to write code that scales to massive amounts of data.

- How to implement data structure?
  - How does it all fit together?

A Thinking & Coding Course

- Coding as well as thinking
  - Learn to think, coding is the tool
  - Requires consistent effort
    - For every lecture, > 2hrs of study time
  - Every lecture builds on the previous one.

Outline

- Course Context & Motivation
- What are Data Structures?
- Course Administrivia
- Data Structures in the Era of Large Language Models
- Importance of Learning Data Structures

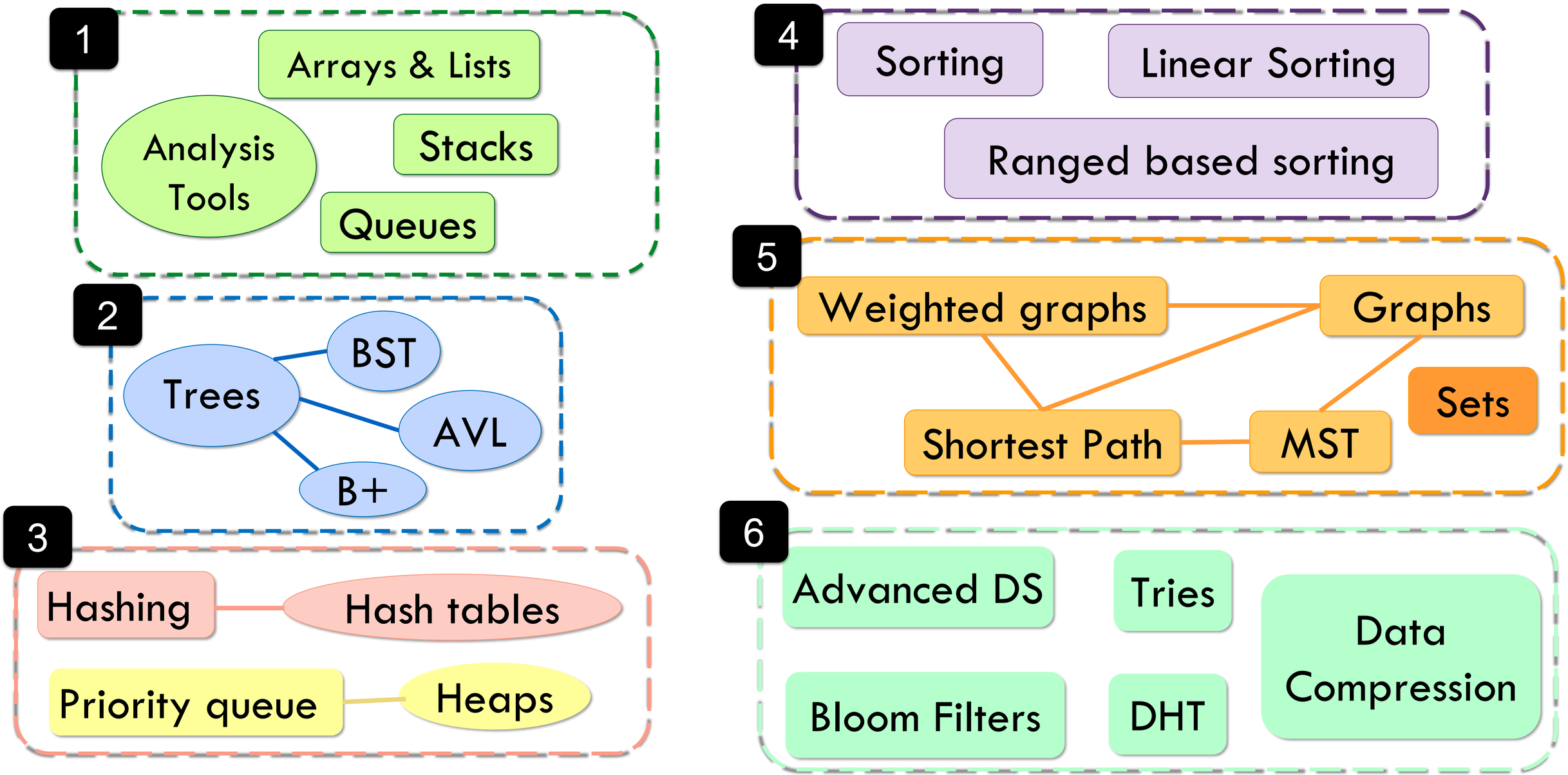
Course Teaching Fellows and Teaching Assistants

1. Muhammad Khaquan – Teaching Fellow
2. Shizza Ashar - Teaching Fellow
3. Abdullah – Head TA (S1)
4. Ali Azhar
5. Umer Irfan
6. Mohammad Ali Wajid
7. Sarfaraz
8. Shahzaib – Head TA(S2)
9. Amna Hassan
10. Areeba Fatima
11. Aleena Abbas
12. Sania Hassan
13. Saifullah Shakir Jan
14. Maryam

Office hours are on LMS as well as Slack

Designated TAs for Assignments and Quizzes

Course Topics and Modules



Course Prerequisites

- CS 200: Introduction to Programming

- We'll assume familiarity with OOP concepts
  - e.g., classes, pointers, inheritance, and templates

- Good to refresh these concepts if you need to

Grading Policy

Assessment	Number	Weightage
Assignments	Three	15%
Quizzes	10 quizzes (n-2) *	20%
Attendance	24 (n-3)	5%
Mid Exam	One 18th Oct, 1:00-4:00pm	25%
Final Exam*	One	35%

***1 ungraded quiz**

*If the university is unable to conduct the final examination due to an emergency, final exam scores will be extrapolated.

Programming Assignments

- 3 Programming Assignments (parts of a system)
  - Individual
  - Programming Language: C++
  - Environment: Linux/Windows/MacOS
  - **Help: TA office hours and Slack**

- Submission Policy
  - You have 5 free late days for the semester to cater to emergencies.
  - However, the free days cannot be used in the dead week

Programming Assignments

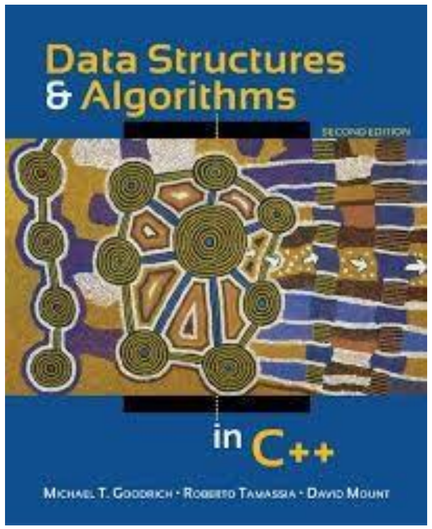
- Very Important component of the course
- Programming can only be learnt by doing
  - Invaluable to spend individual time
  - Learn and unlearn concepts
  - Make mistakes and learn from them
  - Get help only after you have tried
- Everyone can program and it can be very rewarding
  - Your effort now will payoff later

Home works and Quizzes

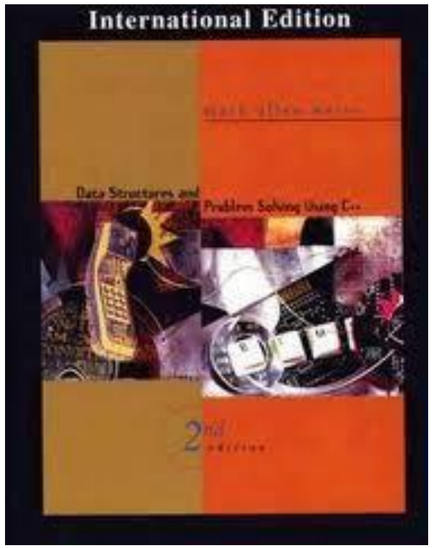
- Five ungraded home works
  - Reinforcement of the concept
  - Prepare for quizzes/ exams/ assignments

- Ten graded announced quizzes with n-2
  - N-2 is to cater emergency situations
  - N-2 policy will not be applied to copied quizzes/assignments

Text Books for the course



- Data Structures and Algorithms in C++ by Michael T. Goodrich, Roberto Tamassia, and David Mount (2nd Edition)



- [Optional] Data Structures and Algorithm Analysis in C++ by Mark Allen Weiss (2nd Edition)

Course Website

- We will be using **LMS** (lms.lums.edu.pk) for communication and sharing files (e.g., slides, quizzes, home works, assignments)

- All discussions/questions about the course, assignments, quizzes, and home works will take place on **Slack**

- Course outline and soft copy of the course textbook is on **LMS**
  - Outline contains readings for each lecture

Course Outline

9	BST Deletion and Balanced BSTs • Deletions in BSTs • What are AVL trees? Height-balance property and analysis of AVL trees	**PA-1 Due 29th September**
10	AVL Trees • Single and double rotations • Impact on insertion and deletion operations	**PA-2 Release 2nd October**
11	B+ Trees • Towards m-ary trees • What are B+ trees? Why do we need them? • B+ tree operations	
Module-3: Hash Tables, Priority Queues, and Heaps		
12	Hash Tables: Hash Functions, Chaining • What are hash tables? What are hash functions? • Hash tables with chaining • Analysis of hash tables	**HW-3 Release (*ungraded*) 9th October**
13	Hash Tables: Open Addressing • Hash tables with open addressing • Linear probing and double hashing	**Course Feedback Survey (1/2)**

Academic Honesty (1/3)

- All coursework must be completed individually & independently
  - You can discuss your understanding of the problem statement and a general sketch of approach
  - You cannot discuss the specific details of solution

- Your Code is your responsibility
  - Never share your solution with anyone, (in this or future semester)
  - Never look at someone else's code, from this or past semesters.

Academic Honesty (2/3)

- You must not use AI to do your assignments
  - This includes generating entire solutions or significant portions of code. If you're unsure, ask the course staff first.

- We use advanced code comparison software
  - Immune to obfuscation
  - Produces color-coded all-student-pairs code comparisons
  - Any academic integrity violation will be reported to the disciplinary committee

Why Integrity Matters – Academic Honesty (3/3)

- Cheating can never be worth it
  - Takes away your opportunity for learning and lowers your confidence
  - You'd never get this time back!
  - Negatively impacts your colleagues
  - Damage to your moral compass

AI Policy – here

1. Maximally Restrictive Policy → Prohibiting AI Use

As per LUMS AI use guidelines, we require that all work submitted for this course be the student's own. The use of generative AI tools (e.g., ChatGPT, Gemini, Llama) is strictly prohibited at all stages of coursework. Any violations of this policy will be considered academic misconduct, regardless of whether they are identified through plagiarism detection tools or other verification methods. Please be aware that AI policies may differ between courses at LUMS, and it is the student's responsibility to meet the specific requirements of each course.

Learning Support for Large-Enrolment Courses

- **Extended office hours** held by TAs and TFs in dedicated consultation/TA rooms to provide individualized or small-group academic support;
- **Tutorials** conducted by TFs to reinforce conceptual understanding and address common challenges;
- **Designated student lounges** that facilitate peer-to-peer learning and informal academic discussions;
- **Online discussion forums** (e.g., slack) enable timely support from peers, TAs, and TFs for resolving pressing academic queries

Together, these components constitute an integral part of the course design and are essential for deepening understanding and enhancing learning outcomes.

Ethics and Data Structure Design

- Design decisions encode values
  - They reveal our assumptions about the world and the people who will be interacting with our design and benefiting from it

Efficient use of resources

Promoting Autonomy

Promoting Neutrality

Priority of the worst-off

- Value tensions and conflicts may arise when the system operates in the world
  - Ex: autocompletions (which suggestions?), priority queue (social assistance programs), options (travel routes on maps)

Agenda

- Course Context & Motivation
- What are Data Structures?
- Course Administrivia
- Data Structures in the Era of Large Language Models
- Importance of Learning Data Structures

AI's Impact on Software Engineering

- AI tools are revolutionizing how we program…
  - … dramatically accelerating development cycles and augmenting developer capabilities

- Key Questions
  - Will AI serve as a replacement or enhancement for developers?
  - Should engineers still master CS fundamentals in an AI-driven world?

Impact of AI

- While LLMs can generate code, understanding how to program and why specific data structures are chosen remains crucial

- LLMs are tools; programmers are the architects and builders

- Programmers guide LLMs, debug their output, and integrate them into larger systems

Why Learn Fundamentals

- Critical Context
  - While LLMs excel at code generation and manipulation, they can produce plausible but incorrect solutions
  - Strong technical foundations are now more crucial than ever - engineers must be able to:

Understand	Knowledge of a language, coding, etc
Verify	Ability to debug, test, analyze, etc
Optimize AI Generated Code	Knowledge of best practices, software optimization

Agenda

- Course Context & Motivation
- What are Data Structures?
- Course Administrivia
- Data Structures in the Era of Large Language Models
- Importance of Learning Data Structures

C++ has a lot of these data structures built-in…

- Why write code and reinvent the wheel?
  - Learning through implementation → to gain deep understanding to make informed design decisions
  - Developing software engineering skills → to master code maintainability, testing and debugging
  - Analysis and experimentation → to systematically optimize software efficiency
  - Innovation and customizations → to develop specialized data structures

Importance of Learning Data Structures

- Your Career's foundation
  - **virtually mandatory in interviews** by major software companies like Google, Meta, and Amazon
  - Building scalable systems
  - The competitive edge

- Language of Research
  - Powering Research

- Essential for advanced courses

Your journey this semester

- What you'll gain:
  - Deep understanding of fundamental data structures
  - Critical thinking skills for evaluation of solutions
  - Performance optimization techniques
  - Problem-solving approaches that transcend specific tools

Questions