Programming and Data Structures

CS2700

Rupesh Nasre.
rupesh@cse.iitm.ac.in

TAs: Abhishek, Kaushal, Lokasis, Shan,
Solarica, Akshay, Keshav

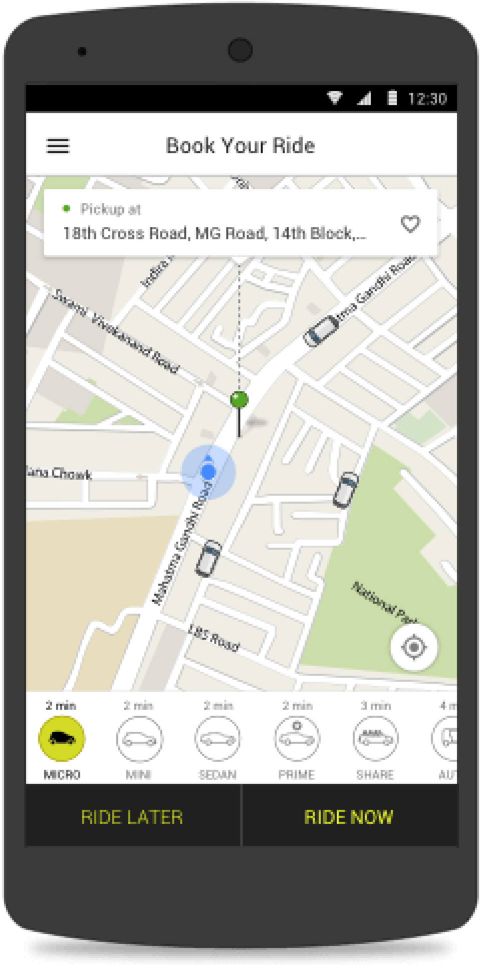
Course webpage: ~rupesh/teaching/pds/aug21
Moodle: https://courses.iitm.ac.in/course/view.php?**id=399**

August 2021

Placement in Computer Science

- CS1100: Coding
- CS1200: Proofs, Counting
- CS2200: Computation Theory
- CS2300: Overview of Digital World
- CS2600: Hardware
- **CS2700: Efficient Implementation**
- CS2800: Algorithms
- CS3100: Ways of Programming
- CS3300: Translation (Programmer and Machine)
- CS3500: Resource Management (User and Machine)

Relevance, with an example

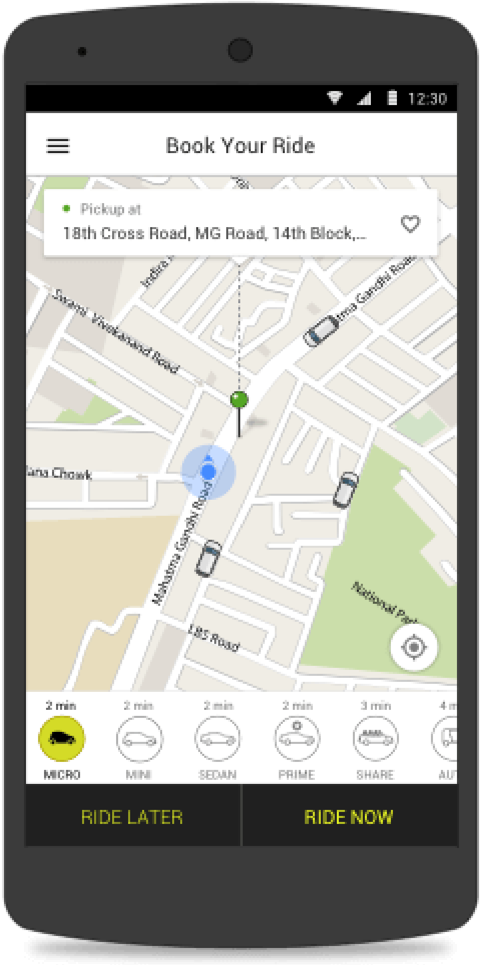


- DM: How many ways can you go from IIT to Central?
- LMC: Can you compute *blah* using a smartphone?
- CO/CA: How to build a smartphone hardware?
- **PDS: How to keep track of cabs such that a passenger can query the nearest cabs efficiently?**
- Algo: How to compute the shortest path from IITM to Central?
- OS: How to give a higher priority to an incoming call?
- Compilers: Translating an app or OS to machine code
- Networks: How does a phone call work?
- DBMS: How to store and retrieve world-map data relevant to the user?

Do not decouple these subjects (especially Algorithms and Data Structures). They go hand-in-hand, but we emphasize on one hand at a time.

Image source: Google images

More Examples



- How to keep track of cabs such that a passenger can query the nearest cabs efficiently?
- How should I store addresses such that I can show suggestions as user types a destination address?
- How should I categorize cabs such that the passenger is able to view Micro, Mini, Prime, Sharing options quickly?
- I should be able to identify quickly if there are multiple sharers nearby.
- When a cab moves, how should I store the data such that the graphics rendering uses only the diff rather than displaying the complete screen again?
- How should I store previous rides such that I am able to find an approximate cost for this journey prior to booking?

Data structures get more important whenever there is more data, … and more types of data.

Image source: Google images

Misconceptions

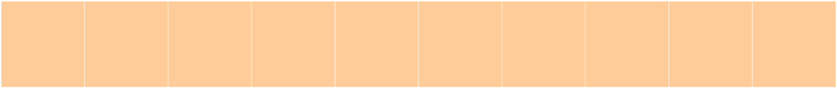
- Data structures get created when we use struct.
- Data structures need pointers.
- An algorithm must use a specific data structure.
- C++ has more data structures than C.
- There are in total seven data structures.
- Union-Find is the best data structure.

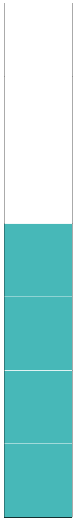
What is it?

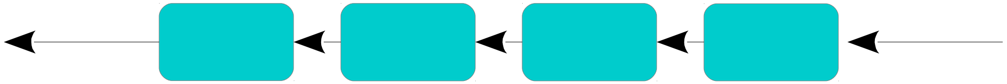
- **Data Structures** is about organizing data such that its storage and retrieval improve the efficiency of the algorithms using it.
- A data structure may be used by multiple algorithms.
- An algorithm may use multiple data structures simultaneously.
- An algorithm may use different data structures and achieve the same computation.

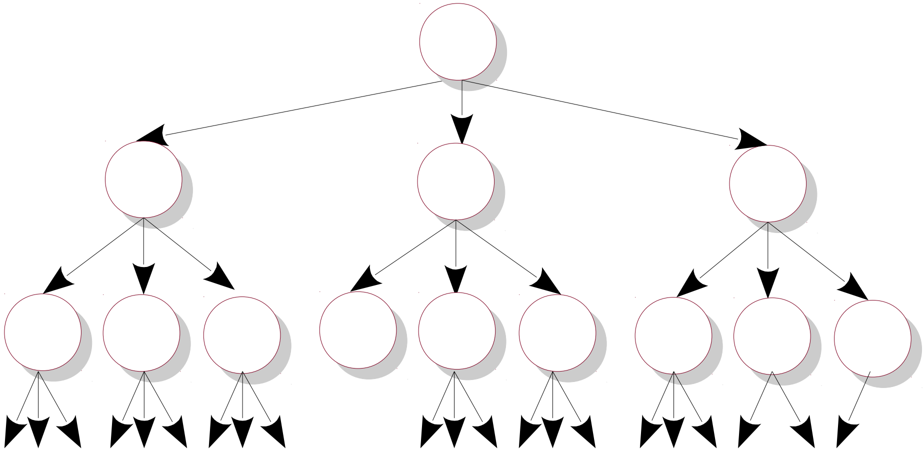
Core and Standard

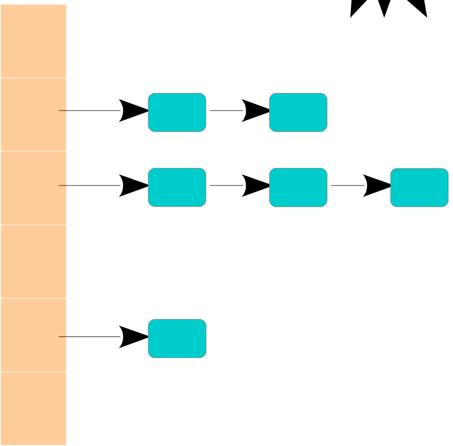
- Array
- Linked List
  - Stack
  - Queue
- Tree
  - Binary Tree
  - Binary Search Tree
  - Heap
  - …
- Hash Table
- Graph

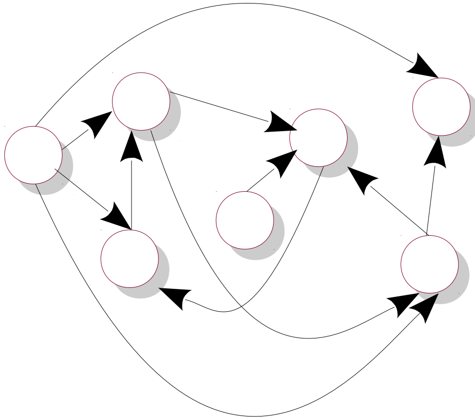












Learning Outcomes

- **Choose efficient data structures and apply them to solve problems.**
- Design correct programs to solve problems.
- Analyze the efficiency of programs based on time complexity.
- ~~Prove the correctness of a program using loop invariants, pre-conditions and post-conditions in programs.~~

Logistics

- Theory: Join here.
  - B slot (Mon 9, Tue 8, Wed 12, Fri 11)
  - Hackerrank submissions
- ~~Lab: DCF~~
  - ~~R slot (Wed 14)~~
- ~~Prerequisite: CS1200 Discrete Mathematics~~
- It is your responsibility to get subscribed to moodle (id=399)
  - Self-enrolment is enabled.

Evaluation

- Assignments
  - Problems on Moodle / Hackerrank
  - Mostly C (a little bit of C++)
  - Plagiarism checker will be used
- Q1, Q2, Q3
  - 20 marks each, 1 hour
  - On Moodle
  - Dates differ by a couple of days from the institute calendar (check course webpage).
  - You have this week to suggest changes.

To get the MOST out of this course

- Keep hands away from mouse, keyboard, and whatsapp.
- Solve questions during classwork.
  - Keep a copy with you. Take notes.
- Ask questions (others also haven’t understood).
  - Do not let a few dominate the discussion.

Programming and Data Structures

CS2700

Rupesh Nasre.
rupesh@cse.iitm.ac.in

TAs: Abhishek, Kaushal, Lokasis, Shan, Solarica, Akshay, Keshav

Course webpage: ~rupesh/teaching/pds/aug21
Moodle: https://courses.iitm.ac.in/course/view.php?**id=399**

August 2021