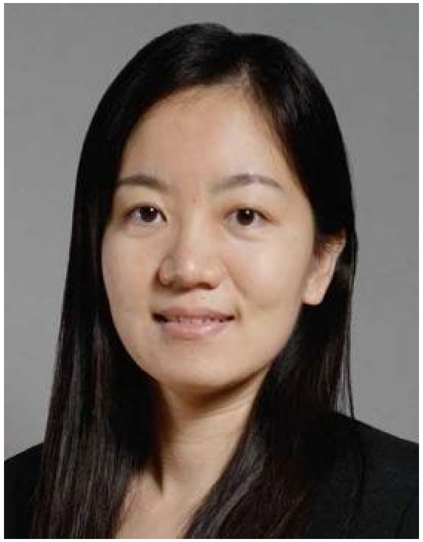
CSD2181/CDD2183 Data Structures Module Overview

Dr Pei Yiyang

Yiyang.pei@singaporetech.edu.sg

Teaching Faculty



Associate Professor
Dr. Pei Yiyang
Yiyang.pei@singaporetech.edu.sg

Week 1-6 Online Sessions
(Mon and Tue)

Associate Professor
Dr. Hua Guang
guang.hua@singaporetech.edu.sg

Week 1-6 Face-to-Face Sessions
(Thu)

Assistant Professor
Dr. Nisha Jain
Nisha.jain@singaporetech.edu.sg

Week 8-13 All Sessions

What to expect from the module?

- Classical **Data Structures**
  - Stacks, queues, trees, graphs, etc.
- Basic **Algorithm Analysis**
  - Complexity representation
- Low-level **Implementation** (C++)
  - Basic data structures and their variations
  - Popular algorithms such as searching and sorting.

Assessment and Tentative Schedule

Assessment	Weightage	Remark	Deadline
Programming Assignments	20%	• 5 assignments with 4% each • 1 bonus assignment with 2%	Week 3, 5, 8, 10, 12, 14
Quizzes	18%	• 3 quizzes with 6% each • LMS quizzes	Week 3, 8, 11
Midterm	20%	• LMS quiz	Week 6
Final	30%	• Format to be defined later	Week 14
In-Class Exercises	12%	• 12 quizzes with 1% each	Weekly

Difficulty level: In-class exercises < Quizzes < Midterm and Final

Class Format

- Online Lectures: 2x1.5 hours per week
- Face-to-face session: 1 hour per week
  - Summary of online lectures
  - In-class exercises (via ClassPoint)
  - Assignment briefing
  - Consultation

Tentative Class Schedule

Week	Lecture (Mon)	Lecture (Tue)	F2F Class (Thu)
1	Module Overview Program Design	Memory Management	• Exercise 1 • Assignment 1 Briefing
2	Algorithm Analysis	Algorithm Analysis	• Exercise 2
3	Sorting	Sorting Quiz 1 (~30 min)	• Exercise 3 • Assignment 2 Briefing
4	Abstract Data Type	Abstract Data Type	• Exercise 4
5	Binary Trees	Binary Search Trees	• Exercise 5
6	AVL and Splay Trees (Pre-recorded videos)	Midterm Test (~1 hour)	• Exercise 6 • Assignment 3 Briefing

Logistics

- SIT’s LMS
  - Announcement
  - Lecture recording
  - Quizzes, midterm (and final)
  - Discussion forum for each topic and assignment
- Digipen’s Moodle
  - Assignment posting and submission

Logistics

- Quizzes, midterm (and final)
  - LMS with lockdown browser and webcam
  - Make sure you go through the trial test to check if your lockdown and webcam are working well before the first quiz in Week 3.

- In-class exercises
  - Conducted using ClassPoint during F2F classes.

- Absence from assessment
  - Must submit a leave application via in4sit.
  - Must inform the instructor on the same day.

- Makeup Tests
  - No makeup for quizzes and in-class exercises. Marks will be based on the average of assessment of the same type, if with valid reason.
  - Will be arranged only for midterm and final, if with valid reason.
  - Will be harder than the original tests.

Introduction to Data Structures

Outline

- Data Structure
- Algorithm
- Their Relationship
- Data Structure Examples
  - Queue vs Stack
  - Array vs Linked List

Data Structure

- A specialized arrangement of data in a computer's memory so that it can be used efficiently.
- Analogy
  - – Container in a kitchen

Algorithm

- A set of step-by-step instructions designed to perform a specific task.
- Analogy
  - Cooking recipe

Pancake recipe
1
2
3
4
5
6
7
8
9

Data Structure and Algorithm

- A data structure provides the foundation for organizing and accessing data efficiently.
- An algorithm uses that data organized in the data structure to perform a specific task or solve a specific problem in a systematic way.
- Analogy
  - Recipes (algorithms) specify the sequence of steps to use the ingredients (data) in the containers (data structure) to make dishes (task).

Data Structure and Algorithm

- The performance of a program to perform a specific task (algorithm) depends on the way in which the data is organized (data structure).
- The choice of a data structure depends on the processes that need to be performed on that data (algorithm).
- Data structures and algorithms are closely related.

What is the module about?

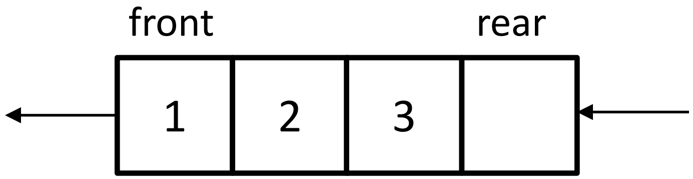
- This course is about the study of several data structures and their use in algorithms.
  - The similarities and differences
  - The pros and cons
  - The trade-off between time and space complexities

Data Structures

- Arrays
- Linked List
- Queues
- Stacks
- Trees (Binary Search Trees, AVL Trees, Splay Trees, Red Black Trees)
- Graphs
- Hash Maps
- Skip Lists
- Heaps

Queues

- Definition
  - Linear data structure
  - First-In-First-Out (FIFO)
  - Elements are added at the rear and removed from the front.
- Characteristics
  - Elements processed in the order they're added

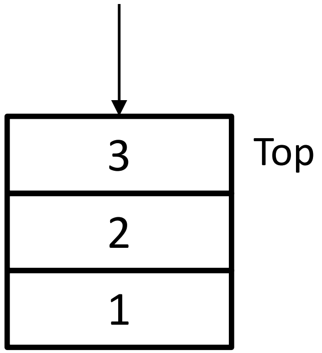


Applications of Queues

- Print jobs in printers
- Task scheduling in operating systems
- Web server handling incoming requests
- Etc

Stacks

- Definition
  - A linear data structure
  - Last-In-First-Out (LIFO)
  - Elements are added and removed from the same end (..e., the top).
- Characteristics
  - Elements processed in reverse order of addition

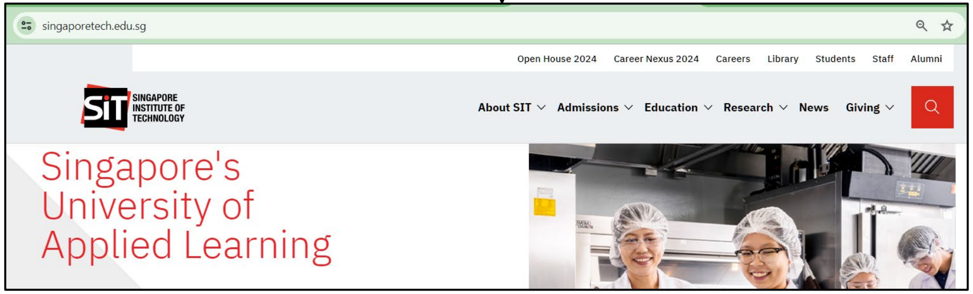


Applications of Stacks

- Back button in web browsers
- Undo/redo functionality in applications
- Function call stack in programming
- Expression evaluation (postfix, infix, prefix)
- Etc.

Applications of Stacks: Back Button in Web browsers

SIT Home Page



singaporetech.edu.sg

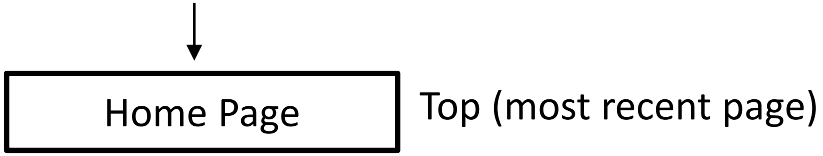
Open House 2024 Career Nexus 2024 Careers Library Students Staff Alumni

SINGAPORE INSTITUTE OF TECHNOLOGY

About SIT Admissions Education Research News Giving

Singapore's University of Applied Learning

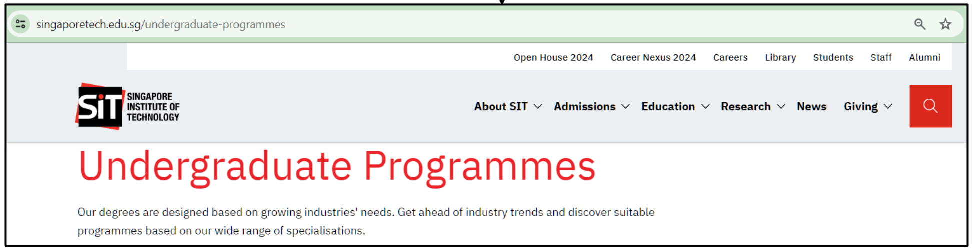
Stack for Storing Browsing History



Home Page

Top (most recent page)

SIT Undergraduate Programmes



singaporetech.edu.sg/undergraduate-programmes

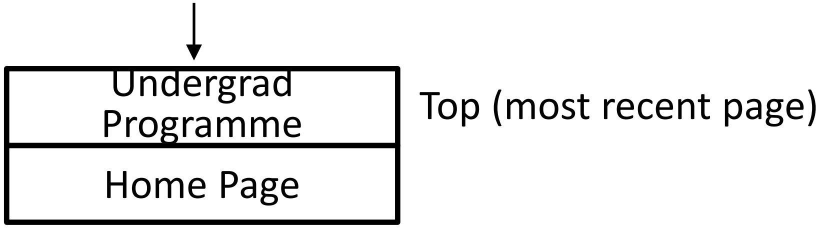
Open House 2024 Career Nexus 2024 Careers Library Students Staff Alumni

SINGAPORE INSTITUTE OF TECHNOLOGY

About SIT Admissions Education Research News Giving

Undergraduate Programmes

Our degrees are designed based on growing industries' needs. Get ahead of industry trends and discover suitable programmes based on our wide range of specialisations.

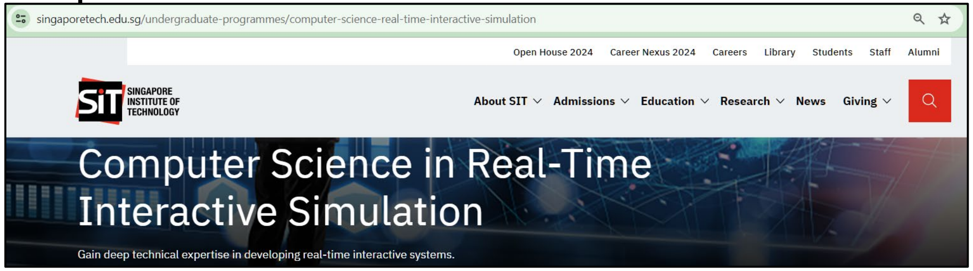


Undergrad Programme

Home Page

Top (most recent page)

SIT-Digipen B.S
Computer Science in Real-Time Interactive Simulation



singaporetech.edu.sg/undergraduate-programmes/computer-science-real-time-interactive-simulation

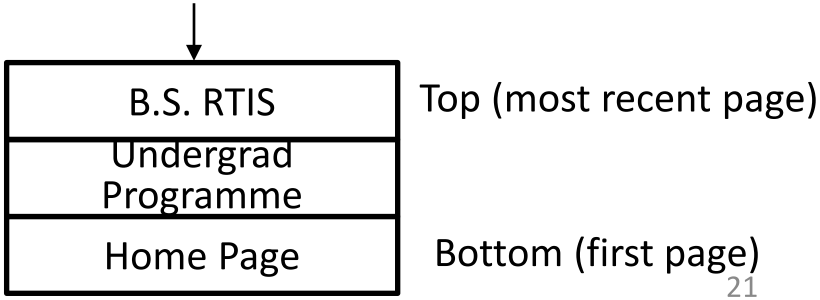
Open House 2024 Career Nexus 2024 Careers Library Students Staff Alumni

SINGAPORE INSTITUTE OF TECHNOLOGY

About SIT Admissions Education Research News Giving

Computer Science in Real-Time Interactive Simulation

Gain deep technical expertise in developing real-time interactive systems.



B.S. RTIS

Undergrad Programme

Home Page

Top (most recent page)

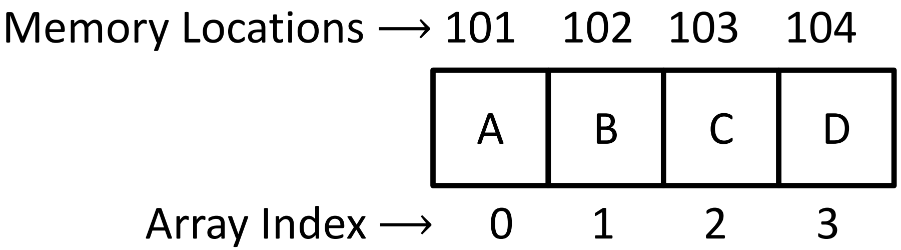
Bottom (first page)

Choosing Data Structures

- The choice of a data structure (e.g., stack or queue) is influenced by the nature of the algorithm.
  - FIFO: Queue is more suitable.
  - LIFO: Stack is more suitable.
- The effectiveness of an algorithm in achieving its intended task (e.g., FIFO) relies on the functionalities offered by the chosen data structure.
  - Queue: More effective
  - Stack: Not effective

Arrays

- Definition:
  - A linear data structure
  - A fixed-sized collection of elements of the same type are stored in contiguous memory locations.
- Characteristics
  - Elements occupy adjacent memory locations
  - Direct access to elements via indices
  - Fixed size



Memory Locations ⟶ 101 102 103 104

A B C D

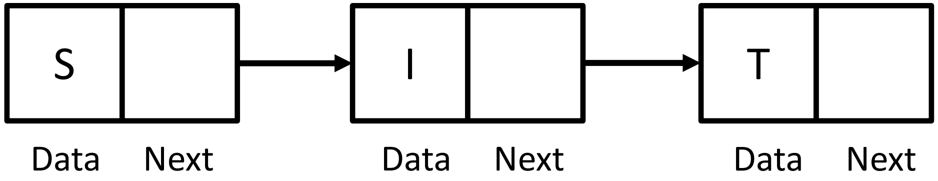
Array Index ⟶ 0 1 2 3

Applications of Arrays

- Pixel values in images
- Vectors and matrices in mathematical computations
- Player scores in games
- Implementation of other data structures such as queues and stacks

Linked List

- Definition:
  - A linear data structure
  - Elements are stored in non-contiguous memory locations, but connected via links to the previous and/or next element(s) (e.g., using pointers).
- Characteristics
  - Efficient insertion and deletion at any position
  - Dynamic size



Applications of Linked Lists

- Playlist Management in Music Applications
- Linked allocation of files
- Free memory blocks by memory managers
- Implementation of other data structures such as queues and stacks

Array vs Linked List

	Array	Linked List
Memory location	Contiguous	Mon-contiguous
Size	Fixed in size	Dynamic in size
Memory allocation	At compile time (Static)	At runtime (Dynamic)
Memory usage	Less memory needed per element	More memory per element due to the need to store both data and links to previous and/or next element(s)
Random access (search)	Faster, by index	Slower, Traversing of the list
Insertion and deletion	Slow	Fast

Choosing Data Structures

- The choice of the data structure (e.g., array or linked list) is influenced by the requirements of the algorithm.
  - For random access: array is more suitable.
  - For dynamic operations: Linked list is more suitable.
- The algorithm's effectiveness in achieving its intended task (e.g., frequent random access) relies on the functionalities offered by the chosen data structure.
  - Array: faster
  - Linked list: slower

Summary

- Introduction to Data Structure
  - Data Structure
  - Algorithm
  - Their Relationship
  - Data Structure Examples
    - Queue vs Stack
    - Array vs Linked List