CSC 212: Data Structures and Abstractions

01: Introduction

Prof. Marco Alvarez

Department of Computer Science and Statistics
University of Rhode Island

Spring 2026

THINK BIG WE DO℠

CSC 212

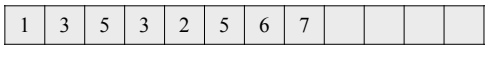
Learn to model and solve complex problems with computers.

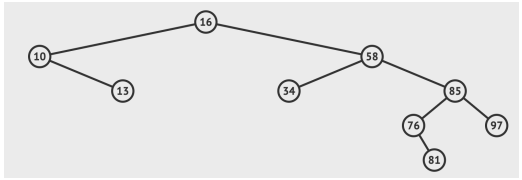
Course description

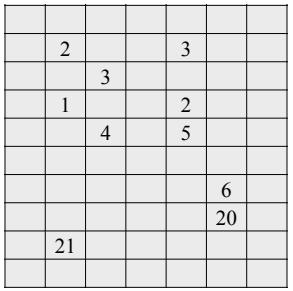
- Introduction to **fundamental data structures** and their algorithms
  - ✓ arrays, lists, stacks, queues, trees, hash tables, graphs (most popular topics for job interview questions)
  - ✓ survey of classic algorithms for sorting and searching
- Basic principles of analysis of algorithms
  - ✓ improve your foundation of CS theory
- Writing code that runs efficiently
  - ✓ choosing good algorithms and data structures

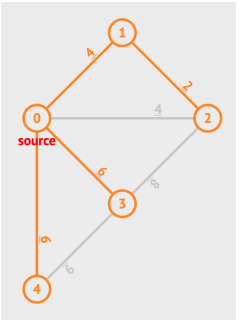
Presumes a strong foundation in C++ programming, including:

- Pointers
- Classes and objects
- Recursion







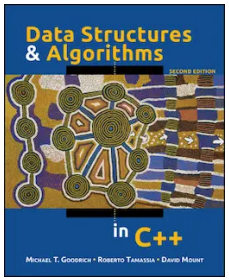


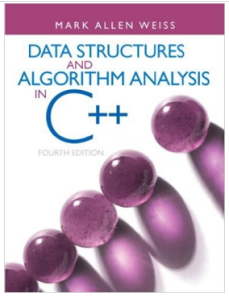
Course organization

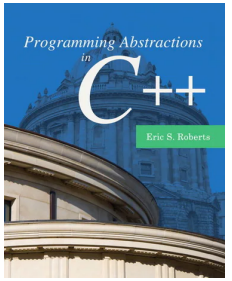
Course information

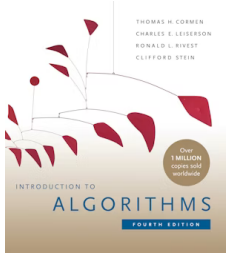
- Lectures
  - ✓ TR 9:30 - 10:45a (Kirk AUD)
- Labs
  - ✓ W 10 - 11:45a (Library 166)
  - ✓ W 12 - 1:45p (Library 166)
  - ✓ W 2 - 3:45p (Tyler 53)
- Course Website
  - ✓ https://homepage.cs.uri.edu/~malvarez/teaching/csc-212/

Recommended textbooks









Support tools

Academic discussion, polls, quizzes.

Assignment submission and grading.

Virtual meetings and office hours.

Coursework

- Homework assignments
  - ✓ individual work, however discussions and collaboration are allowed
    - **you must write your own code** and solutions to problem sets
  - ✓ late submissions NOT accepted
    - ample time given to complete (6-8 days)
    - start early and use office hours for guidance and feedback
- Exams
  - ✓ in-person
  - ✓ **no electronic devices allowed**
  - ✓ mix of multiple-choice, and short-answer questions designed to test understanding

Academic integrity

- Assignments
  - ✓ each student/team must submit their own **unique** solutions, sharing/copying solutions from peers is **prohibited**
- AI and LLMs
  - ✓ AI tools (e.g., ChatGPT, Gemini, Claude, GitHub Copilot) can be used to enhance learning through brainstorming, concept exploration, and strategy development
    - students must critically evaluate and fully understand any AI-generated content used in their work
    - all AI-assisted work **must be cited in submissions**
  - ✓ AI tools are designed to support students' learning, NOT to replace independent problem-solving and critical thinking

Grading (subject to change)

- Assignments (25%)
  - ✓ programming
  - ✓ problem sets
- Exams
  - ✓ midterm 1 (20%)
  - ✓ midterm 2 (25%)
  - ✓ final exam (25%)
- Attendance (5%)
  - ✓ lectures
  - ✓ labs

Average of all exams ≥ 50 is mandatory to pass, regardless of final weighted grade

What is expected from you?

- Attend lectures/labs
  - ✓ students are expected to attend all lectures and labs
  - ✓ regular attendance is linked to higher grades and better comprehension of course material
- Participate and think critically
  - ✓ ask questions (lectures, labs, office hours, Ed, …)
- Start working on assignments early
  - ✓ refrain from simply copying and pasting answers generated by LLMs
- Laptops and cellphones are **NOT permitted** unless being used for taking notes

Resources

Need a refresher on C++ programming?

- Pick a textbook (learn syntax)
- Solve Challenges

Kattis

CODEFORCES
Sponsored by TON

codewars

topcoder™

LeetCode

Warming up

- Adjacent elements sum
  - ✓ find the **maximum sum** of any pair of **adjacent** elements in an array of integers

14

11	3	5	3	2	-5	6	7	-9	-2	13	2

first pair

Practice

- Complete the following program:

```cpp
#include <vector>
#include <iostream>

int max_sum(std::vector<int>& nums) {
    // TODO
    return -1;
}

int main() {
    std::vector<int> nums = {1, -2, 3, 4, -1, 2, 1, -5, 4};
    int result = max_sum(nums);
    std::cout << "Maximum sum is: " << result << std::endl;

    return 0;
}
```