2023-2024 GSMST CS Club Algorithmic Programming Division Syllabus

Anish Goyal (Chief of Competitions)
Kevin Lee (Algorithmic Programming Head)
Shivali Singh (Algorithmic Programming Assistant Head)

Table of Contents

1 Overview

1.1 About Us

Welcome to the 2023-2024 syllabus for GSMST CS Club's Algorithmic Programming Division! Our mission is to nurture the next generation of algorithmic problem solvers at GSMST. We aim to offer a comprehensive learning experience that encompasses various aspects of algorithmic programming, equipping our members with the knowledge and skills needed to excel in solving complex computational challenges and thrive in competitive coding competitions.

1.2 What We Do

1. **Weekly Meetings**
2. **Weekly Problem Sets**
3. **Regional and National Competitions**

1.3 Competitive Opportunities

- Lockheed Martin's CodeQuest
- HP CodeWars
- American Computer Science League (ACSL)
- TeamsCode
- UC Berkeley CALICO
- United States of America Computing Olympiad
- Stanford ProCo
- ...and much more!

2 Weekly Meeting Dates

2.1 Semester I

2.1.1 Introduction to Algorithmic Programming (08/31/2023)

2.1.1.1 Meeting Summary

We will introduce the purpose and format of the Algorithmic programming seminars, briefly going over the major competitions that we will compete in during the school year. We will also introduce the concept of how programming competitions work, and the importance of algorithms in them. We could also bring up past achievements, like those from the LM CodeQuest competition last year. Afterwards, we will go over the various resources that students can use throughout the year. Keep in mind that the esteemed Dr. Hong will be teaching most, if not all, of the content-based seminars.

2.1.1.2 Meeting At-A-Glance

- Introducing the comps: CodeQuest, CodeWars, ACSL
- Go over how competitions work and show some examples of problems
- Past achievements
- Go over resources : Websites, Books, LeetCode, *etc.*

2.1.2 Computer Number Systems (09/07/2023)

2.1.2.1 Meeting Summary

In this seminar, we will learn about computer number systems and how to convert between bases, doing practice in conversion and performing arithmetic in numbers of different bases. We will also mention usage in RGB coloring as well as the ASCII and binary values of characters. The students will then try to solve related ACSL problems to polish their number system knowledge and to familiarize with the ACSL format. If we have some time remaining, we could also go over basic bitwise theory.

2.1.2.2 Meeting At-A-Glance

- Go over Computer Number Systems
- Review worksheets of number system problems from ACSL material

2.1.3 Recursive Functions (09/14/2023)

2.1.3.1 Meeting Summary

For this meeting, we will talk about the significance of recursive functions, using example problems to guide the lesson. The students will split up into groups and try to implement recursive functions to simple questions that the instructors will present. We will also learn about the ACSL pseudo-code and look at practice problems.

2.1.3.2 Meeting At-A-Glance

- Review Recursive Functions
- Split into groups to solve simple recursion problem
- What Does This Program Do?

2.1.4 Greedy Algorithms (09/21/2023)

2.1.4.1 Meeting Summary

This seminar will be mainly about greedy algorithms. We will begin by introducing the main theory and a trivial example where a greedy algorithm is used. Then, we will use practice problems to guide the rest of the lesson. The students will also make their own LeetCode accounts and solve related questions on the platform near the end of the lesson.

2.1.4.2 Meeting At-A-Glance

- Talk about when the greedy algorithm should be used
- Discuss how to know if the greedy algorithm is accurate
- Make LeetCode accounts and solve problems

2.1.5 Time and Space Complexity (09/28/2023)

2.1.5.1 Meeting Summary

We will introduce time and space complexity of algorithms, and why they are important, especially in competitive programming. We will talk about how to recognize the time and space complexities of an algorithm, and how the size of the input can give the direction of what the time complexity of the algorithm should be. We will also discuss comparison-based sorting algorithms, as well as the various searching algorithms.

2.1.5.2 Meeting At-A-Glance

- Analyzing time and space complexity
- Implementing binary search manually and using standard library
- Practicing questions on LeetCode

2.1.6 Miscellaneous Algorithms (10/12/2023)

2.1.6.1 Meeting Summary

For this seminar, we will discuss various miscellaneous algorithms like prefix sums, sliding window, and two pointer, directed especially at problems related to arrays or strings. We want them to have a deeper understanding in how these algorithms can simplify a lot of problems

2.1.6.2 Meeting At-A-Glance

- Prefix Sums/Sliding Window/Two Pointer algorithms

2.1.7 Dynamic Programming (10/26/2023)

2.1.7.1 Meeting Summary

This seminar will focus on dynamic programming due to its usage in a large range of intermediate and advanced programming problems. After studying the importance of recursion and memoization in dynamic programming, we will look at examples like knapsack and subsequence problems.

2.1.7.2 Meeting At-A-Glance

- Understand dynamic programming

2.1.8 Dynamic Programming Continued (11/02/2023)

2.1.8.1 Meeting Summary

We will continue the discussion on dynamic programming, and also further our understanding, like using multi-dimensional arrays in dynamic programming as well. For the remaining time, we will review the topics for the first ACSL contest.

2.1.8.2 Meeting At-A-Glance

- Continuation of dynamic programming
- Review for ACSL contest

2.1.9 Prefix/Infix/Postfix Notation (11/09/2023)

2.1.9.1 Meeting Summary

We will look at prefix/infix/postfix notation, and how to systematically convert between 2 notations. We will discuss the significance of them in computer science, and will look at practice problems from ACSL regarding this topic.

2.1.9.2 Meeting At-A-Glance

- Prefix/Infix/Postfix Notation

2.1.10 FSAs and Regular Expressions (11/16/2023)

2.1.10.1 Meeting Summary

In preparation for the second ACSL competition, we will look at FSAs and Regular Expressions. We will study the various syntax, and do practice with regex libraries on programming languages. The students will divide into groups and compete to come up with correct regex filters first.

2.1.10.2 Meeting At-A-Glance

- FSAs, Regular Expressions
- Students split into groups to compete and write regex filters for problems stated by officers

2.1.11 LISP (11/17/23)

2.1.11.1 Meeting Summary

For this seminar, we will review FSAs and Regular Expressions, and also study the basic syntax for LISP. We will practice with the study material from ACSL.

2.1.11.2 Meeting At-A-Glance

- LISP

2.2 Semester II

2.2.1 Data Structures (01/18/2024)

2.2.1.1 Meeting Summary

As our first seminar in the second semester, we will study various data structures often used in programming competitions, like vectors, stacks, and queues. We will discuss the differences and advantages between those data structures, and look at practice problems where certain data structures are more applicable. We will also talk about the upcoming competitions in second semester, like CodeQuest andCodeWars Preparation

2.2.1.2 Meeting At-A-Glance

- Data Structures
- Review of vectors/arrays, stacks, queues
- Discussion on the dates and format of CodeQuest, CodeWars, CALICO, etc.

2.2.2 Bitwise Operations (01/25/2024)

2.2.2.1 Meeting Summary

We will study how bitwise operations are powerful in competitive programming to decrease the difficulty and time complexity of algorithms, and how many problems are bitwise operations in disguise. We will also talk about using the associativity and commutativity of bitwise operations to our advantage

2.2.2.2 Meeting At-A-Glance

- Bitwise Operations
- Practice problems

2.2.3 Introduction to Graph Theory (02/01/2024)

2.2.3.1 Meeting Summary

In this seminar, we will introduce the basic terminology of graph theory, and how many problems can be transformed into graphs and solved with graph theory knowledge. We will talk about the differences between DFS and BFS, as well as the implementations.

2.2.3.2 Meeting At-A-Glance

- Basic graph theory terminology and techniques
- DFS and/or BFS

2.2.4 Graph Theory With Applications (02/08/2024)

2.2.4.1 Meeting Summary

This meeting will be a continuation of the graph theory lesson. We will review the DFS and BFS, as well as extend those algorithms into Dijkstra's algorithm. Most of the seminar will be spent on solving basic practice questions in graph theory.

2.2.4.2 Meeting At-A-Glance

- More graph theory
- Review on DFS/BFS, Dijkstra's algorithm

2.2.5 String Methods (02/22/2024)

2.2.5.1 Meeting Summary

With competitions on the horizon and due to how competitions like CodeQuest and CodeWars comprise of a lot of string-based problems, we will talk about the techniques used in string processing, like the utilization of hashmaps, algorithms like two-pointer and sliding window, and applications of dynamic programming to give students an idea on how to approach string-based problems.

2.2.5.2 Meeting At-A-Glance

- String processing: Common algorithms/methods involving strings

2.2.6 CodeWars Preparation (02/29/2024)

2.2.6.1 Meeting Summary

We will review the logistics and teams for CodeWars, and solve practice problems from past CodeWars competitions.

2.2.6.2 Meeting At-A-Glance

- CodeWars Prep & Logistics

2.2.7 Logic Gates and Assembly (03/14/2024)

2.2.7.1 Meeting Summary

As the final seminar where we will be learning something new, we will finish off with learning about logic gates and basics of the Assembly language. We will have worksheets for students to practice logic gates, and students will write basic Assembly code to become familiar enough for the ACSL competitions.

2.2.7.2 Meeting At-A-Glance

- Digital electronics and Assembly language

2.2.8 In-House Programming Competition (03/21/2024)

2.2.8.1 Meeting Summary

Instead of a traditional meeting, we will be hosting our in-house programming competition. More information regarding registration and teams will be released later.

2.2.8.2 Meeting At-A-Glance

- School-wide algorithmic programming competition

2.2.9 Competition Prep (03/28/2024)

2.2.9.1 Meeting Summary

Due to upcoming competitions (UC Berkeley CALICO and TeamsCode), we will solve various past problems in those competitions and will review logistics.

2.2.9.2 Meeting At-A-Glance

- UC Berkeley CALICO Prep
- TeamsCode Prep

2.2.10 Mini Practice Contest (04/11/2024)

2.2.10.1 Meeting Summary

For final practice before upcoming competitions, we will have a mini practice contest where students will compete with each other to solve various problems based on the topics they've learned throughout the year. This is supposed to be more of a fun event to allow students to become more familiar and de-stress.

2.2.10.2 Meeting At-A-Glance

- Fun practice competition for members to unwind

2.2.11 Final Competition Prep (04/18/2024-04/25/2024)

2.2.11.1 Meetings Summarized

We will be having final prep before the competitions (CodeQuest and/or Stanford ProCo), with Question and Answer sessions about the logistics of the competitions and algorithms that we've taught through the year. There will also be a list of practice competitions that the students can work through.

2.2.11.2 Meetings At-A-Glance

- Final prep before CodeQuest and Stanford ProCo (subject to actual competition dates/availability)