GSMST Algorithmic Programming Yearlong Proposal

Anish Goyal Kevin Lee Shivali Singh

Table of Contents

1 List of Competitions

Here is a list of competitions that GSMST Algorithmic Programming will participate in for the entire school year with their competition dates and any team/school information requirements.

1.1 Lockheed Martin's Code Quest

- Two options: in-person and virtual
  - Schools can send up to three teams for the virtual event.
    - A "Team" consists of 2-3 people and a coach/mentor.
  - Schools can send unlimited teams for the in-person event.
- Free registration; information TBD
- Novice and advanced divisions
- Competition takes place in late April

1.2 HP CodeWars

Honestly, this seems like a copy of Lockheed Martin's Code Quest. I don't know which competition came first, but we should register nonetheless, since it is a great opportunity:

- Each school needs to have a sponsor.
  - Schools can only have one sponsor.
  - A sponsor can manage up to three teams.
    - Teams can only have 2-3 people.
- Novice and advanced divisions
- In-person and virtual options
- Takes place on March 2, 2024

1.3 ACSL

The American Computer Science League offers programming problems and short (MCQ) questions. There are five divisions based on skill level. It is the only competition for Algo that will **cost money** this year.

- 4 Contests
  - First competition is in late October with a **2-3 month window** for each contest.

- The cost to register a team in a division is $150 and $75 for each additional team.
- You can have up to 12 students per team
  - The problems themselves are completed individually, however (think picoCTF), with points being assigned to each team member.
  - Each team has a 5-score and a 3-score
    - This means that the team score for each contest is the sum of the top 5 or top 3 student scores in that contest.

1.4 TeamsCode

An online platform with two annual competitions: one in the Spring and the other in the Summer. This competition is school-agnostic, meaning they don't care about the number of teams per school.

- Summer contest is in early August
- Spring contest is in early April
- Teams can have up to four students
- Novice and advanced divisions
- Free registration

1.5 UC Berkeley CALICO

- Up to three people per team
- School agnostic
- Approximately happens during the second Saturday of April
- Free registration

1.6 Stanford ProCo

- **Requires travel to Stanford**
  - First come first serve basis, with acceptance into the competition confirmed beforehand
- Up to three people per team
- Novice and advanced divisions
- Occurs in mid April

2 Algo Syllabus

Kevin, Anish, and Shivali (the GOAT of all time) have already talked about what we want to cover on the syllabus for the entire year, but none of us have the leisure to start drafting a syllabus at this time (we cba), so we will probably work on it towards the end of July. Therefore, the deadline for the syllabus will be **August 1**.

2.1 Syllabus Research

Here is the syllabus research that we discussed as a collective during our meeting on June 3 at 9:01 PM. Approximately *66.66666%* of the meeting participants came prepared with three topics to cover for the syllabus and why those topics were important for future competitions and the broader aspect of cybersecurity. All we have to do now is figure out when we want to cover these topics and how we could structure our lesson plans for them, which we'll do in another meeting towards the end of July. A general **disclaimer**: It's possible that not all of our ideas and scheduled seminars will get used since Mr. Hong will be teaching on some of those days.

2.1.1 Anish Goyal's Research

- **Sorting Algorithms**
  - Understanding various sorting algorithms, such as bubble sort, insertion sort, and quicksort, is essential as it allows efficient organization and arrangement of data, optimizing search and retrieval operations in applications. (Try not to make this a repeat of CSA?).
- **Graph Algorithms**
  - Exploring graph algorithms like breadth-first search (BFS) and depth-first search (DFS) enables the analysis and manipulation of interconnected data structures, which is crucial for tasks like network analysis, pathfinding, and recommendation systems.
- **Dynamic Programming**
  - Studying dynamic programming techniques aids in solving complex optimization problems by breaking them down into smaller overlapping subproblems, leading to improved efficiency and faster computation in fields like computational biology, resource allocation, and scheduling. (Ok we technically covered this last year but its important so maybe we should again?)

2.1.2 Kevin Lee's Research

- **Binary Search/Other searching algorithms**
  - Sometimes binary search is just the solution or the bottleneck to entire problems, so knowing the logic behind it would enhance efficiency. We would also need to consider various cases behind it to avoid off-by-one errors and save time. Studying searching algorithms with sorting algorithms would help improve algorithmic intuition
- **Recursion**
  - Crucial to understand virtually every other topic like dynamic programming and graph theory and will improve understanding of logic. We can also studying backtracking through recursion which can help us transition into dynamic programming.
- **Miscellaneous algorithms like Sliding Window/Two Pointer:**
  - Although covered last year, using these methods can help solve various elementary/intermediate algorithmic problems. Also helps improve algorithmic intuition.

2.1.3 Shivali Singh's Research

For these suggestions, I was wondering if we could cover the math/algorithms behind machine learning—if this doesn't work then I can put in other things:

- **Linear regression**
  - Linear regression is a supervised learning algorithm that is used to predict a continuous value from a set of input features. The math behind linear regression is based on linear algebra. In linear regression, the goal is to find a line that best fits the data. The line is represented by a set of coefficients, which are found using a process called least squares. Least squares is a method for finding the line that minimizes the sum of the squared errors between the line and the data points.
- **Logistic regression**
  - Logistic regression is a supervised learning algorithm that is used to predict a categorical value from a set of input features. The math behind logistic regression is based on probability and statistics. In logistic regression, the goal is to find a probability model that can be used to predict the probability of a data point belonging to a particular category. The probability model is found using a process called maximum likelihood estimation. Maximum likelihood estimation is a method for finding the parametters of a probability model that maximize the likelihood of the data.
- **Decision trees**

– Decision trees are a supervised learning algorithm that is used to classify data into different categories. The math behind decision trees is based on graph theory. In decision trees, the data is represented as a tree, where each node in the tree represents a decision that needs to be made. The leaves of the tree represent the different categories that the data can be classified into. The decision tree is created using a process called recursive partitioning. Recursive partitioning is a method for dividing the data into smaller and smaller subsets until each subset contains only data points of the same category.

3 Seminar Dates

We have decided to keep seminar dates on Thursdays out of necessity. This is because Thursday is the only day that Mr. Hong is available to mentor us for ACSL, so we just defaulted to that day.

4 Budgeting

- $95 for an ACSL team.
  - If we garner enough members, we can register a second team for $75 extra = $160.
  - We'll basically charge members a flat fee to cover the cost. Since each team can have up to 12 people, we would charge each person a one-time fee of $10 to participate in ACSL. So that gives us a margin of error of about 6 people we can "lose" for that second team and still break even.
- Yeah everything else is free so hip hip hooray!

5 Conclusion

Algorithmic programming is gonna be a success this year. We could work out a field trip to Stanford for *Stanford ProCo*, but the budget for that is TBD since it would require cooperation on everyone's part. We will work hard to ensure that we can make our seminars/workshops more interactive and fun for everyone and prepare everyone for the competitions that we will have this year.