Planeación Taller

**Objetivo**: que los chavos puedan enfrentarse solos a problemas y puedan resolverlos.

Algoritmos + Implementación

Siempre pensar en la cajita de herramientas

**Itinerario de actividades**

Todas las clases se transmiten en vivo y se quedan grabadas.

En periodos de dos semanas (asumiento lunes y miércoles):

* Lunes - clase tema: dudas, ejemplos, problemita compartido con tarea
* Miércoles - clase problemas: 3 problemas en equipo o solo, dos diferentes a la tarea y uno igual
* Lunes - clase tema: dudas, ejemplos, problemita compartido con tarea
* Miércoles - concurso: 5 problemas: 3 de los temas, 2 radoms
* Tarea de tres problemas a la semana: básico, intermedio, avanzado

Se guarda lista de progreso en un google sheets.

* Editorial todos los lunes (solución y código de los problemas de tarea y de las sesiones de problemas)

Los problemas estarán basados en: leetcode, codeforces y omegaUp.

**Temas**

Es una lista extensible, y obviamente no se puede cubrir todo. La idea es que se vayan eligiendo los temas entre todos, aunque igual sí hay unos temas que es importante ver antes que otros.

* Introducción a super básica a C++ y correr programitas
* Sintaxis básica de C++:
  + Lectura y escritura
  + Tipos de datos: int, string, char, float, double
  + Arreglos y arreglos dinámicos
  + Ifs
  + Loops
  + Sort de c++
* Dividir en casos
* Complejidad computacional
* Two pointers / Sliding window
* Búsqueda binaria
* Intro a más estructuras de datos: conjuntos, hash tables, colas, pilas, heaps
* Usar tu propia función de sort
* Backtracking
* Intro a DFS en matrices
* Intro a BFS en matrices
* Programación dinámica – DP
* Greedy algorithms
* Intro a grafos
* DFS y BFS en grafos
* Manipulación de BITs
* Algoritmos famosos de colas y pilas
* Divide and Conquer
* Binary Indexed Trees
* Segment Trees-
* Algoritmos de grafos: Kruskal, Djikstra, Topological, Floyd-Warshall, Bellman-Ford
* String algorithms
* Union find
* Sparse Tables
* Tries