

Act 5.2 - Actividad Integral sobre el uso de códigos hash

Este código refleja la esencia de la eficiencia algorítmica en el contexto de un concurso de programación. A través de una estructura de datos bien diseñada, como el vector de concursantes y la estructura Concursante, el programa procesa de manera eficiente la información de los participantes.

La lectura de datos y el procesamiento de intentos se realizan de manera lineal, lo que demuestra una eficiencia notable al evitar operaciones innecesarias. La elección de estructuras de datos como map y set para rastrear intentos incorrectos y problemas resueltos, respectivamente, muestra una consideración cuidadosa para optimizar la búsqueda y la inserción.

La fase de filtrado de concursantes válidos también se maneja de manera eficiente, evitando la inclusión de concursantes no relevantes. La complejidad del filtrado es constante, contribuyendo a la eficiencia general del programa.

Sin embargo, la joya de la optimización reside en la etapa de ordenamiento. La complejidad $O(N \log N)$ aquí es la principal fuente de carga computacional. Aunque es inevitable en un proceso de clasificación, el impacto se minimiza al limitar el conjunto de datos a los concursantes válidos.

En resumen, este código no solo resuelve el problema en cuestión sino que también lo hace de manera eficiente. La atención a la elección de estructuras de datos y la optimización del proceso de ordenamiento son evidentes. Esto destaca la importancia de la eficiencia algorítmica, especialmente en escenarios donde la cantidad de datos puede afectar significativamente el rendimiento.

Link del código en Github: [TC1031-Portafolio_Final-/main.cpp at main · ArturoL0709/TC1031-Portafolio_Final- \(github.com\)](https://github.com/ArturoL0709/TC1031-Portafolio_Final-/main.cpp)