

Programación con GPUs

Práctica 01 - Introducción

Segundo Semestre, 2025

ATENCIÓN CON LAS PRÁCTICAS

En programación, la práctica es un componente fundamental del aprendizaje. Hacer estas prácticas te van a dar la oportunidad de poder aplicar lo aprendido en la teoría, entender mejor los conceptos y probar tus habilidades.

La recomendación de la cátedra es que:

- **Leas MUY bien el enunciado:** Analices el enunciado y entiendas exactamente lo que hay que hacer. Parece sencillo, pero es común resolver un ejercicio diferente al planteado.
 - **No busques soluciones óptimas inmediatamente:** Primero intentá resolver el problema y luego pensá si la solución es óptima.
 - **Dediques tiempo a pensar:** No te desesperes si no se te ocurre la solución inmediatamente. Es común que las soluciones no salgan a la primera. Pensá en el problema, quizás volvé a leer la teoría y fijate si se te ocurre. Son problemas complejos y a veces hay que darles tiempo para que se asienten.
 - **¡No busques soluciones rápidas!:** No busques rápido en internet la solución o vayas a leer la solución a la guía de resoluciones inmediatamente. Cada solución que leas rápido te va a dar la falsa sensación de comprensión y te va a sacar la posibilidad de tener el momento **¡AHA!** donde realmente entendiste cómo resolver un problema. **¡Te entendemos, es difícil a veces!**, pero es parte del proceso de aprendizaje.
 - **¡No te desanimes!:** Si volviste a pensarlo un tiempo y no se te ocurre nada, es momento de dejar el problema por un tiempo y retomarlo luego.
 - **¡Volví al problema luego de un tiempo y no me sale!:** Si volviste a pensar el problema y no se te ocurre nada, es momento de leer la solución. No te sientas mal por esto, pero cuando te sientes a leer la solución. El proceso de leer la solución implica *entenderla*, y NO copiarla. Una vez que entiendas la solución, esperá un tiempo para escribirla y probarla.
 - **¿Y si no entiendo la solución?:** Si no entendiste la solución, anotá las dudas, tratá de pensar qué es lo que te falta y preguntá a los docentes de la cátedra. ¡Nunca te quedes con la duda!
 - **ChatGPT (cualquier LLM) lo resuelve todo:** ¡Es verdad!, pero como cualquier herramienta, cuanta más teoría sepamos, mejor podremos utilizarla.
- ¡Suerte en la práctica!**

Contents

1.1	¿Qué es escalabilidad vertical? ¿por qué tiene un límite?	3
1.2	¿Qué es la escalabilidad horizontal? ¿Cómo mejora la escalabilidad vertical? . . .	3
1.3	¿Por qué es importante medir la complejidad algorítmica?	3
1.4	¿Qué es la complejidad en espacio?	3
1.5	¿Qué es la complejidad en tiempo?	3
1.6	¿Por qué no se utilizan GPUs y programación paralela para todo?	3
1.7	¿Qué es la Ley de Amdahl?	3
1.8	¿A qué se denomina <i>Nick's Class</i> ? ¿por qué es importante en la programación paralela?	3

-
- 1.1 ¿Qué es escalabilidad vertical? ¿por qué tiene un límite?
 - 1.2 ¿Qué es la escalabilidad horizontal? ¿Cómo mejora la escalabilidad vertical?
 - 1.3 ¿Por qué es importante medir la complejidad algorítmica?
 - 1.4 ¿Qué es la complejidad en espacio?
 - 1.5 ¿Qué es la complejidad en tiempo?
 - 1.6 ¿Por qué no se utilizan GPUs y programación paralela para todo?
 - 1.7 ¿Qué es la Ley de Amdahl?
 - 1.8 ¿A qué se denomina *Nick's Class*? ¿por qué es importante en la programación paralela?