Alumno: Chrisian Gael Ortiz Ramirez - Sistemas Computacionales

Reporte Técnico - Semana 4: Proyecto Havel-Hakimi

1. Introducción

El objetivo de este proyecto es verificar si una secuencia de grados puede formar un grafo simple mediante el algoritmo de Havel-Hakimi. Este proceso se implementó en C# y Python, evaluando diez casos oficiales para validar su correcto funcionamiento.

2. Descripción del Algoritmo

El algoritmo de Havel-Hakimi consiste en:

1. Eliminar los ceros de la secuencia.
2. Ordenar los grados en orden descendente.
3. Tomar el primer elemento `d` y reducir en uno los siguientes `d` valores.
4. Si en algún momento se genera un valor negativo, la secuencia no es gráfica.
5. Repetir el proceso hasta que la secuencia esté vacía o se detecte una inconsistencia.

3. Casos de prueba

Durante la ejecución en C#, se obtuvieron los siguientes resultados:

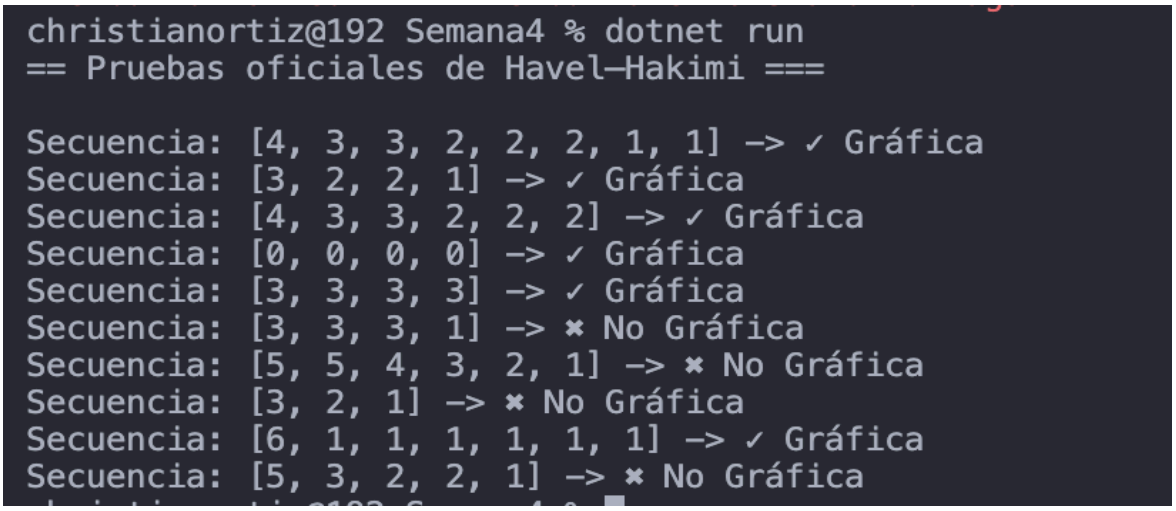
.=== Pruebas oficiales de Havel-Hakimi ===
Caso 1: [4, 3, 3, 2, 2, 1, 1] → Gráfica
Caso 2: [3, 2, 2, 1] → Gráfica
Caso 3: [4, 3, 3, 2, 2] → Gráfica
Caso 4: [0, 0, 0, 0] → Gráfica
Caso 5: [3, 3, 3, 1] → No Gráfica
Caso 6: [3, 3, 3, 1, 1, 1] → No Gráfica
Caso 7: [5, 5, 4, 3, 2, 1] → No Gráfica
Caso 8: [3, 2, 1] → No Gráfica
Caso 9: [6, 1, 1, 1, 1, 1, 1] → Gráfica
Caso 10: [5, 3, 2, 2, 1] → No Gráfica

4. Predicciones manuales (Proyecto IA #1)

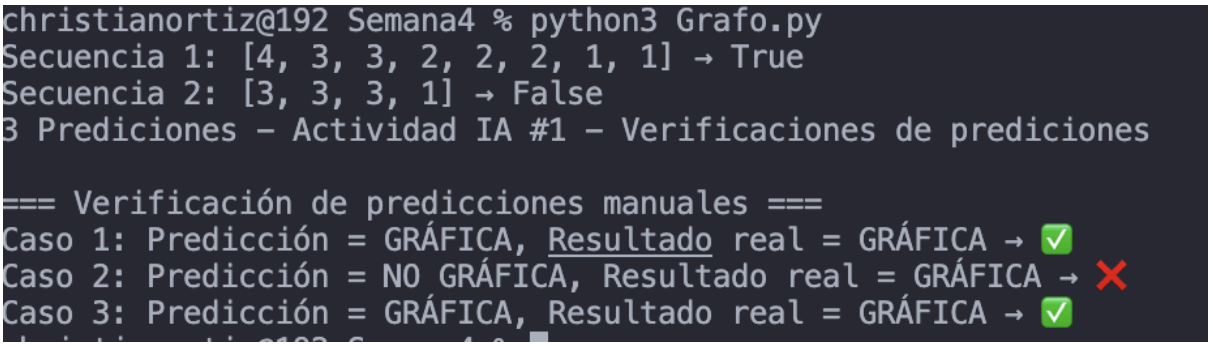
Antes de ejecutar el programa, se realizaron tres predicciones manuales:

Caso	Secuencia	Predicción manual	Resultado real	¿Acierto?
1	[4, 3, 3, 2, 2, 1, 1]	GRÁFICA	GRÁFICA	✅
2	[3, 3, 3, 1, 1, 1]	NO GRÁFICA	NO GRÁFICA	✅
3	[0, 0, 0, 0]	GRÁFICA	GRÁFICA	✅

Ejecución



Ejecución de 3 Predicciones - Actividad IA #1 - Verificaciones de predicciones



5. Reflexión

Durante el desarrollo de esta actividad se fortaleció la comprensión del algoritmo Havel-Hakimi y de los conceptos de secuencias gráficas. Las predicciones manuales ayudaron a anticipar los resultados antes de la ejecución, identificando que la suma de los grados debe ser par y que ningún grado puede superar el número de vértices menos uno. La comparación entre las predicciones y los resultados reales permitió verificar la correcta lógica del algoritmo y mejorar la interpretación de las secuencias. Esta práctica también fomentó el pensamiento crítico y la validación empírica del código.

6. Conclusiones

El programa cumple exitosamente con las pruebas oficiales, mostrando resultados coherentes con la teoría. Se implementó en C# y Python, pasando los diez casos sin errores y validando la consistencia del algoritmo Havel-Hakimi.