

Reporte Consolidado: Semanas 1-8

Estructuras de Datos Avanzadas - 2025

Semana 1: Recursividad vs Iteración

Torres de Hanoi

- **Implementación:** Recursiva $O(2^n)$ vs Iterativa $O(2^n)$
- **Resultado:** Recursiva más rápida (418ms vs 1059ms para n=20)
- **Mejora DP:** 15,000x más rápido con memoización

Escaleras (DP)

- **Problema:** Saltos de 1, 2, 3 escalones
 - **Resultado:** Tabulación 10x más rápida que memoización
 - **n=35:** 1,132,436,852 formas en 0.7ms
-

Semana 2: Programación Dinámica

Laboratorio Completo (5 Ejercicios)

1. **Detección de Patrones:** Tribonacci Sí necesita DP, Factorial NO
2. **Transformación Guiada:** Recursión → Memo → Tabla
3. **Cambio de Monedas:** Monedas [1,3,4], n=6 → 2 monedas (3+3)
4. **Debugging DP:** Errores comunes (tamaño array, casos base, orden)
5. **Salto de Ranas:** 3 implementaciones (Memo, Tabla, O(1))

Resultados Clave

- **Mejora:** $O(2^n) \rightarrow O(n)$
 - **Ahorro:** ~15,000x más rápido para n=30
-

Semana 3: Teoría de Grafos

Mapa de Tráfico Urbano

- **Vértices:** 8 intersecciones (A-H)
- **Aristas:** 12 calles (7 bidireccionales + 5 unidireccionales)
- **Representación:** Listas de adyacencia

Análisis

- **Densidad:** 33.9% (grafo disperso)
- **Ahorro de memoria:** 70.3% (19 vs 64 espacios)
- **Vértice más conectado:** A (Centro Comercial, grado 5)

Semana 4: Havel-Hakimi

Validación de Secuencias Gráficas

- **Algoritmo:** Havel-Hakimi $O(n^2 \log n)$
- **Casos de prueba:** 10/10 pasados
- **Ejemplos:**
 - $[4, 3, 3, 2, 2, 2, 1, 1] \rightarrow \checkmark$ Gráfica
 - $[3, 3, 3, 1] \rightarrow X$ No Gráfica (grado negativo)

Implementación

- C# y Python con trazas detalladas
 - Validación de consistencia (suma par, max grado)
-

Estadísticas Generales

Métrica	Valor
Semanas completadas	4 de 8
Archivos creados	25+
Líneas de código	~3,000
Líneas de documentación	~2,000
Lenguajes	C#, Python, Markdown

Conceptos Dominados

Algoritmos

- Recursión y casos base
- Programación Dinámica (Top-Down y Bottom-Up)
- Grafos (representaciones y propiedades)
- Validación de secuencias (Havel-Hakimi)

Estructuras de Datos

- Listas de adyacencia
- Matrices de adyacencia
- Diccionarios para memoización
- Arrays dinámicos para tabulación

Análisis de Complejidad

- Temporal: $O(1), O(n), O(n \log n), O(n^2), O(2^n)$
- Espacial: $O(1), O(n), O(n^2)$

- Trade-offs: Memoria vs Tiempo
-

📁 Estructura de Archivos

```
HTML de las semanas/
└── Semana 1/
    ├── TorresDeHanoi/
    │   ├── Program.cs
    │   ├── Reporte.md
    │   └── results.txt
    └── Escaleras/
        ├── Program.cs
        ├── Reporte.md
        └── results.txt
└── Semana 2/
    └── Semana2_LabDP/
        ├── Ejercicio1_DeteccionPatrones.py
        ├── Ejercicio2_TransformacionGuiada.py
        ├── Ejercicio3_CambioMonedas.py
        ├── Ejercicio4_DebuggingDP.py
        ├── Ejercicio5_MiniProyecto_SaltoRanas.py
        ├── Reporte_Semana2.md
        └── README.md
└── Semana 3/
    └── Semana3_Grafos/
        ├── Program.cs
        ├── analyze_graph.py
        ├── edges_undirected.txt
        ├── edges_directed.txt
        ├── Reporte_Semana3.md
        └── README.md
└── Semana 4/
    └── Semana4_HavelHakimi/
        ├── Program.cs
        ├── havel_hakimi.py
        └── Reporte_Semana4.md (este archivo)
```

⌚ Próximas Semanas (5-8)

Semana 5: BFS y DFS

- Recorridos de grafos
- Detección de ciclos
- Componentes conexas

Semana 6: Caminos Más Cortos

- Dijkstra

- Bellman-Ford
- Floyd-Warshall

Semana 7: Árboles

- Árboles binarios de búsqueda
- AVL y balanceo
- Recorridos

Semana 8: Proyecto Final

- Integración de todos los conceptos
 - Aplicación práctica completa
-

Fecha: Diciembre 2025

Curso: Estructuras de Datos Avanzadas

Progreso: 50% completado