



UNIVERSIDAD DE BUENOS AIRES  
FACULTAD DE INGENIERÍA

Guía de Ejercicios Arboles

**Algoritmos y Estructura de datos**

**Curso Ing. Gustavo Schmidt**

Aquí tienen una lista de ejercicios para practicar con **árboles Heap** (de máximo y mínimo), organizados en tres niveles de dificultad y agrupados por operaciones como inserción, eliminación y cálculo de complejidades.

### Nivel Básico

#### Heap de Máximo

1. Inserta los valores: **20, 15, 30, 10, 5, 25, 35** en un heap de máximo y muestra el árbol final después de cada inserción.
2. Inserta la secuencia: **40, 35, 50, 10, 5** en un heap de máximo. Explica cómo se reorganiza el heap tras cada inserción.
3. Elimina el valor máximo de un heap con los valores: **50, 40, 30, 20, 10**. Explica cómo se reorganiza el árbol.
4. Construye un heap de máximo a partir de la lista: **[8, 4, 7, 1, 3, 6, 2]** usando el algoritmo de construcción (Heapify).
5. Calcula la cantidad de niveles en un heap de máximo con **15 elementos**.

#### Heap de Mínimo

6. Inserta los valores: **10, 15, 5, 20, 25** en un heap de mínimo y muestra el árbol después de cada inserción.
  7. Inserta la secuencia: **8, 6, 7, 5, 3, 2, 1** en un heap de mínimo. Explica cómo se reorganiza el heap tras cada inserción.
  8. Elimina el valor mínimo de un heap con los valores: **1, 2, 3, 4, 5, 6**. Muestra el estado del heap tras cada eliminación.
  9. Construye un heap de mínimo a partir de la lista: **[12, 5, 8, 1, 3, 6]**.
  10. Calcula el valor mínimo de un heap de mínimo sin modificar su estructura.
- 

### Nivel Intermedio

#### Heap de Máximo

11. Inserta una secuencia de **30 valores aleatorios** en un heap de máximo y explica cómo queda el árbol al final.
12. Elimina el máximo valor de un heap con **30 elementos**. Realiza al menos tres eliminaciones consecutivas y analiza el proceso.
13. Construye un heap de máximo a partir de la lista: **[40, 25, 15, 30, 50, 45, 10]**. Utiliza el método de "Heapify desde abajo".
14. Implementa un algoritmo para encontrar el **segundo valor máximo**

en un heap de máximo y calcula su complejidad.

15. Resuelve cuántas comparaciones se realizan al insertar **n elementos** en un heap de máximo.

### Heap de Mínimo

16. Inserta una secuencia de **30 valores aleatorios** en un heap de mínimo. Explica cómo cambia la estructura tras cada inserción.
17. Elimina el mínimo valor de un heap con **25 elementos**. Realiza al menos cinco eliminaciones consecutivas y analiza los cambios.
18. Construye un heap de mínimo usando "Heapify desde arriba" con la lista: **[18, 12, 20, 25, 10, 5, 30]**.
19. Calcula el número de intercambios que se realizan al construir un heap de mínimo con **50 elementos**.
20. Determina cuántos niveles tiene un heap de mínimo con exactamente **127 elementos**.
- 

### Nivel Avanzado

#### Heap de Máximo

21. Inserta los valores: **[100, 90, 80, 70, 60, 50, 40, 30, 20, 10]** en un heap de máximo. Luego, elimina el valor máximo hasta que el heap esté vacío.
22. Escribe un algoritmo para encontrar el **k-ésimo valor más grande** en un heap de máximo y analiza su complejidad.
23. Compara la construcción de un heap mediante inserciones sucesivas frente al uso de "Heapify". Justifica cuál es más eficiente para **1,000 elementos**.
24. Diseña un algoritmo para fusionar dos heaps de máximo en uno solo. Explica el costo de tiempo de esta operación.
25. Analiza el peor caso de complejidad de eliminar el valor máximo de un heap de máximo con **nnn elementos**.

#### Heap de Mínimo

26. Inserta los valores: **[50, 40, 30, 20, 10, 5, 1]** en un heap de mínimo. Luego, elimina el valor mínimo hasta que el heap esté vacío.
27. Implementa un algoritmo para buscar un rango de valores (por ejemplo, todos los elementos entre 10 y 50) en un heap de mínimo. Calcula su complejidad.
28. Diseña un algoritmo para verificar si un árbol binario cumple las propiedades de un heap de mínimo.

29. Fusiona dos heaps de mínimo con  $nnn$  y  $mmm$  elementos respectivamente. Analiza la complejidad del proceso.
30. Analiza cuántos intercambios se realizan en promedio al construir un heap de mínimo con **1,000 elementos**.