

Video29 (1)

Conceptos Principales

- 1. a
- 2. b
- 3. c

Notas

1. Concepto general de eliminación en un heap

- **Siempre se elimina la raíz:**
 - Min-heap → raíz tiene el valor mínimo.
 - Max-heap → raíz tiene el valor máximo.
 - Propósito: **ordenamiento**, no búsquedas rápidas.
 - Al eliminar la raíz, se reemplaza con el **último elemento del árbol** (nivel más bajo, a la derecha) y luego se ajusta el heap.
-

2. Ajuste del heap (heapify hacia abajo)

- **Proceso (min-heap):**
 1. Comparar el nuevo valor en la raíz con sus dos hijos.
 2. Seleccionar el **mínimo de los dos hijos**.
 3. Si el hijo seleccionado es menor que la raíz, hacer **swap**.
 4. Repetir el proceso hasta que el nodo esté en la posición correcta.
 - **Ejemplo paso a paso:**
 - Eliminar **7** de la raíz:
 1. Reemplazar con **50** (último elemento).
 2. Comparar **50** con hijos **12** y **31** → swap con **12**.
 3. Comparar **50** con nuevos hijos → swap con el mínimo.
 4. Repetir hasta que **50** esté en posición correcta.
 - Este procedimiento se repite para cada eliminación sucesiva en la raíz.
-

3. Implementación en arreglo

- El heap se mantiene en un **arreglo lineal**, usando fórmulas:
 - Hijo izquierdo de **i**: $2*i + 1$
 - Hijo derecho de **i**: $2*i + 2$
 - Padre de **i**: $\text{floor}((i - 1)/2)$
 - **Ventaja:** Solo se mueven valores, no punteros ni memoria dinámica.
-

4. Propiedades importantes

- Árbol siempre **completo**: niveles llenos de izquierda a derecha.
- Cada eliminación genera un **nuevo valor de raíz**, manteniendo la propiedad de min/max-heap.
- La lista de elementos eliminados se construye en **orden ascendente** (min-heap) o descendente (max-heap).
- **Búsqueda no eficiente**: si necesitas buscar un elemento específico, requiere búsqueda lineal.