

Video28 (1)

Conceptos Principales

1. a
2. b
3. c

Notas

1. Inserción en un heap (árbol de montículo)

- **Concepto general:**
 - Los nuevos elementos se insertan siempre en el **último nivel**, lo más a la **derecha posible**, para mantener el **árbol completo**.
 - Luego se ajusta el árbol para mantener la propiedad de **min-heap o max-heap**.
-

2. Ajuste del heap (heapify hacia arriba)

- **Proceso:**
 1. Comparar el nuevo elemento con su **padre**.
 2. Si viola la propiedad del heap:
 - **Min-heap:** Intercambiar si el hijo < padre.
 - **Max-heap:** Intercambiar si el hijo > padre.
 3. Repetir hasta que el elemento llegue a su posición correcta (posiblemente la raíz).
 - **Ejemplo práctico:** Insertar 5 en un min-heap:
 - Comparar con el padre 16 → swap → 5 sube.
 - Comparar con el siguiente padre 15 → swap → 5 sube.
 - Comparar con la raíz 7 → no swap → final.
-

3. Implementación en arreglo

- **Fórmulas para localizar padres e hijos:**
 - Padre de posición i : $\text{floor}((i - 1) / 2)$
 - Hijo izquierdo de i : $2*i + 1$
 - Hijo derecho de i : $2*i + 2$
 - **Ventaja:** No se necesita mover punteros ni gestionar memoria dinámica, solo mover valores dentro del arreglo.
-

4. Principios clave

- Mantener siempre el **árbol completo**.
- Las inserciones son **rápidas** porque se coloca el elemento al final y luego se ajusta.

- El heap puede ser **min-heap** o **max-heap**, dependiendo de la prioridad de los elementos.