

Semana 2: Colas de prioridad

1. Colas de prioridad

Se atienden según la prioridad de cada elemento. Cada elemento funciona como un "map", con un par valor-prioridad.

1.1 Tipos de colas de prioridad

- **Cola de prioridad de mínimos:**

El elemento más pequeño se atiende primero.

Ejemplo: 1, 2, 3, 4, 5, 6, 7, 8

- **Cola de prioridad de máximos:**

El elemento más grande se atiende primero.

Ejemplo: 10, 5, 3, 2, 1

1.2 Operaciones principales

1. Crear una cola de prioridad vacía.
 2. Insertar un elemento: `void push(T const& e)`.
 3. Consultar el elemento más prioritario: `T const& top() const`.
 4. Eliminar el primer elemento: `void pop()`.
 5. Determinar si la cola de prioridad está vacía: `bool empty() const`.
 6. Consultar el número de elementos: `int size() const`.
-

1.3 Propiedades de árboles binarios

- Un árbol binario de altura h es **completo** si:
 - Todos sus nodos no finales tienen hijo izquierdo y derecho.
 - Todas sus hojas están en el nivel h .
 - Un árbol binario es **semicompleto** si es completo o tiene vacantes consecutivas en el nivel h empezando por la derecha.
-

1.4 Propiedades matemáticas

1. Un árbol binario completo de altura $h \geq 1$ tiene 2^{i-1} nodos en el nivel i , para $1 \leq i \leq h$.
 2. Un árbol binario completo de altura $h \geq 1$ tiene $2^h - 1$ hojas.
 3. Un árbol binario completo de altura $h \geq 0$ tiene $2^{h+1} - 1$ nodos.
 4. La altura de un árbol binario semicompleto con n nodos es $\lfloor \log(n) \rfloor + 1$.
-

2. Montículos binarios

Un **montículo binario de mínimos** es un árbol binario semicompleto donde:

- El elemento raíz es menor o igual que todos sus hijos (izquierdo y derecho).
-

2.1 Ejemplo de inserción (mínimo)

Cuando insertamos un nuevo elemento:

1. Recorremos el árbol desde la última posición derecha hacia arriba.
2. Si encontramos un elemento mayor que el nuevo, lo sustituimos.
3. Repetimos el proceso hasta que el nuevo elemento esté en su posición correcta.

Ejemplo: Insertar el 14 en el montículo.

Se busca la posición correcta recorriendo el árbol.

3. Costes de operaciones

1. **Eliminar el elemento más prioritario:** $O(\log N)$.
2. **Insertar un nuevo elemento:** $O(\log N)$.
3. **Colocar una lista ordenada en un montículo:** $O(N)$ si ya está ordenada.