

1. Cola de Prioridad

El orden de salida no es el de entrada, sino que está dado por una prioridad. Video YouTube [6] y Video YouTube Ejercicios [7].

Heaps: Propiedad de heap (de máximos) Un árbol binario AB tiene propiedad de heap si la raíz es más grande (o igual) que sus dos hijos, y estos también son heaps.

Heaps: Representación de árbol izquierdista Un árbol izquierdista es aquel que tiene todos los niveles completos, salvo a lo sumo el último, que debe completarse se izquierda a derecha.

Heaps: Representación en arreglos Árbol izquierdista con propiedad de heap \rightarrow En un arreglo como si fuera el recorrido por niveles del AB.

La implementación de una cola de prioridad se hace a través de un arreglo pero se usa la representación de árbol como interpretación.

1.1. Heap: Representación en arreglo - Posiciones

Si estoy en la posición i :

- Posiciones de sus hijos:
 - hijo izquierdo: $2i + 1$
 - hijo derecho: $2i + 2$
- Posición de su padre: $\frac{i-1}{2}$

1.2. Operaciones sobre Heaps: Upheap y Downheap

Encolar:

- Al encolar, guardamos el elemento al final.
- Aplicamos **Upheap**:
 - Si el elemento es más grande que su padre, entonces se intercambian, y se aplica Upheap al padre.
 - Sino, termina.

Desencolar:

- Llevamos el último al primer lugar (podemos intercambiar)
- Aplicamos **Downheap**:

- Si alguno de los hijos es más grande que el elemento, entonces intercambiar, con el mas más grande de ellos.
- Sino, termina.

Complejidad:

- Upheap: $O(\log(n))$
- Downheap: $O(\log(n))$
- Ver el máximo: $O(1)$
- Encolar: $O(\log(n))$
- Desencolar: $O(\log(n))$

1.3. Heapify: Cómo construir un heap desde un arreglo

Si tengo un arreglo de n elementos, cómo creo un heap con dichos elementos.

Complejidad: $O(n)$

1.4. Heapsort

El método requiere dos etapas. Video YouTube [\[8\]](#).

1. Construir un heap con los elementos del arreglo.
 - heap de maximos si queremos ordenar de menor a mayor (creciente).
 - heap de minimos si queremos ordenar de mayor a menor (decreciente).
2. Desencolar los elementos del heap. Se lleva a cabo el intercambio del primer elemento (posición inicial del heap) con el último, y la reconstrucción del heap en una zona que se disminuye en uno en cada etapa. Esta parte concluye cuando la zona del heap queda reducida a 1.

Complejidad: Se termina haciendo n veces Downheap (desde la raíz siempre)
 $O(n \cdot \log(n))$