

Estructuras de datos

Herramientas de programación para
procesamiento de señales

Índice

- Estructuras de datos abstractas
- Cola de prioridad
- Binary heap

Estructuras de datos abstractas

- Abstracción formal
 - Definición de operaciones
- Tipos:
 - Listas: ordenadas, encadenadas (simple, doble)
 - Cola: operaciones en los extremos, FIFO, etc (stack)
 - Mapa: clave, objeto
 - set
- Implementación
- Niveles de jerarquía

Cola de prioridad

- Definición:
 - Cola con un valor asignado a cada elemento (prioridad)
- Operaciones:
 - insert
 - remove_top
 - Change
 - get_top
- Implementación:
 - binary heap

Estructuras de datos abstractas

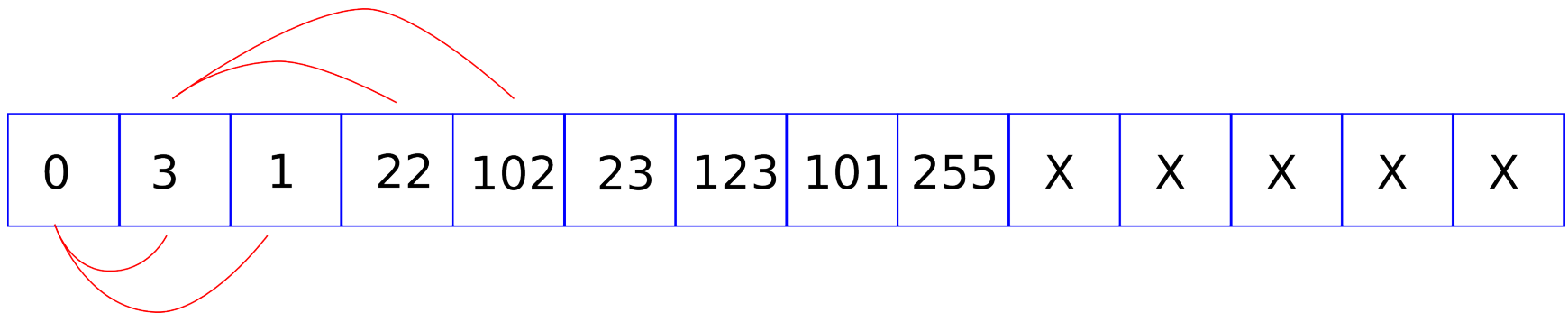
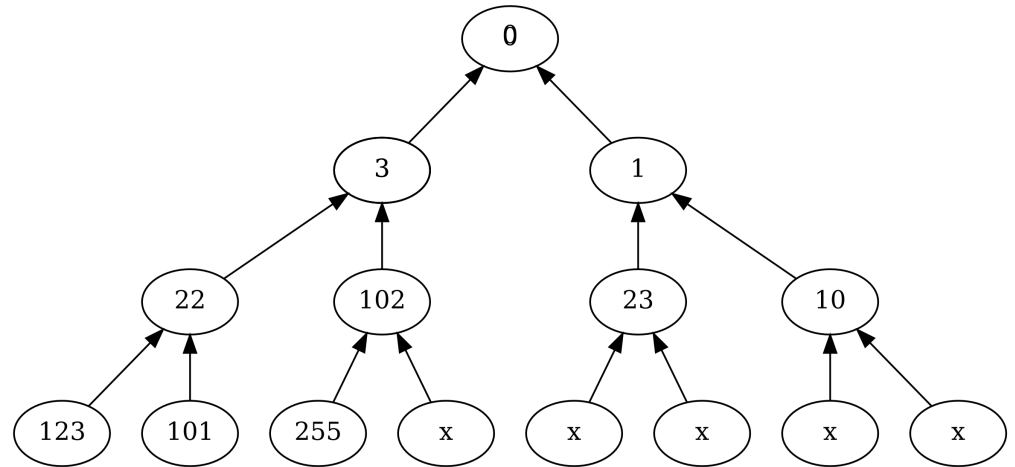
	insert	delete	search	find max	r. access	almac.
u. array	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(n)$
u. linked list	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
heap	$O(\log n)$	$O(\log n)$	$O(n)$	$O(1)$	X	$O(n)$

Heap (1)

- Definición:
 - árbol binario
 - Shape property
 - Heap property
- Datos:
 - Cualquier estructura
 - Comparación: predicado binario
 - Tipos: max-min

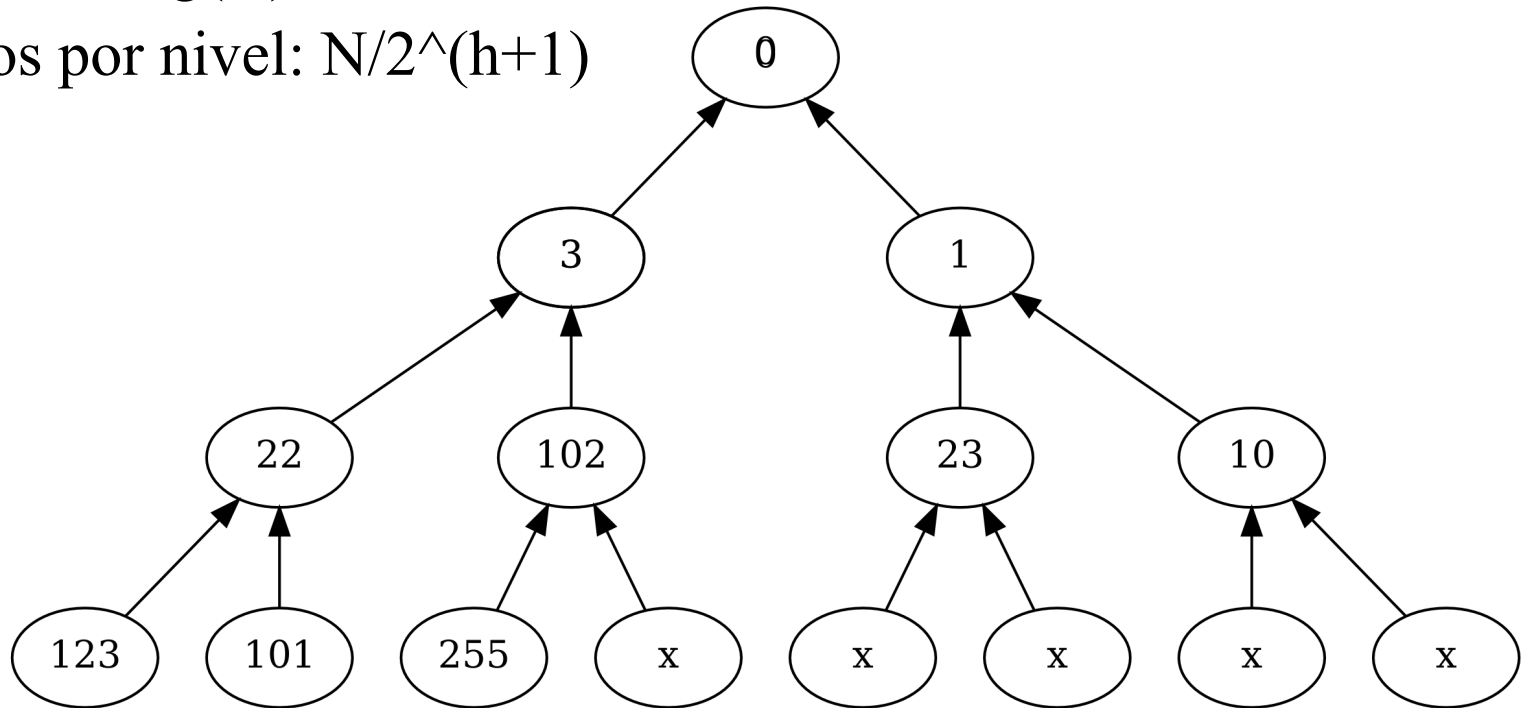
Heap (2)

- Almacenamiento:
 - array de largo fijo
- Operaciones de árbol:
 - Parent: $\text{floor}((i-1)/2)$
 - Left child: $2i+1$
 - Right child: $2i+2$



Heap (3)

- Árbol binario balanceado
 - Hojas: $N/2$
 - Internos: $N/2$
 - Niveles: $\log(N)$
 - Nodos por nivel: $N/2^{(h+1)}$

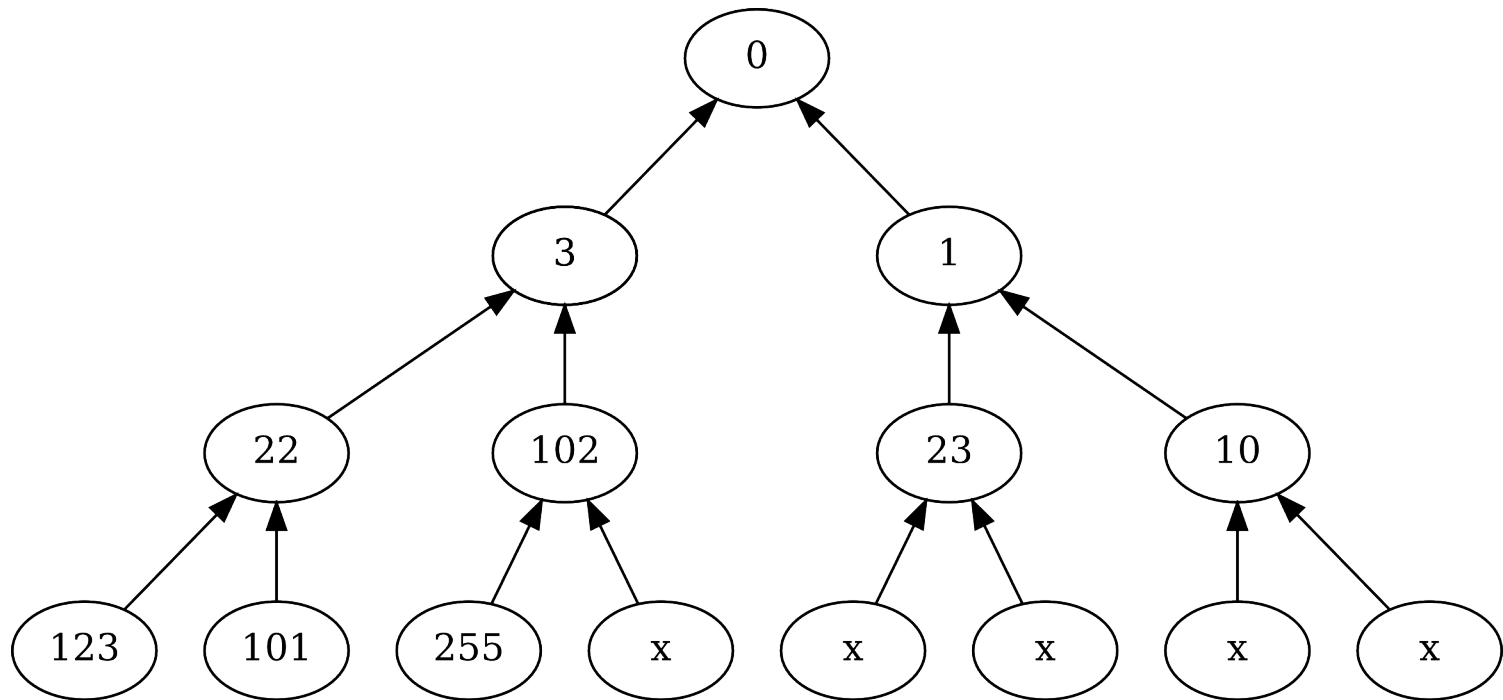


Heap (4)

- Operaciones:
 - publicas:
 - isEmpty()
 - insert()
 - removeTop()
 - **changePriority()**
 - internas
 - moveUp()
 - moveDown()
 - left(), right(), parent()

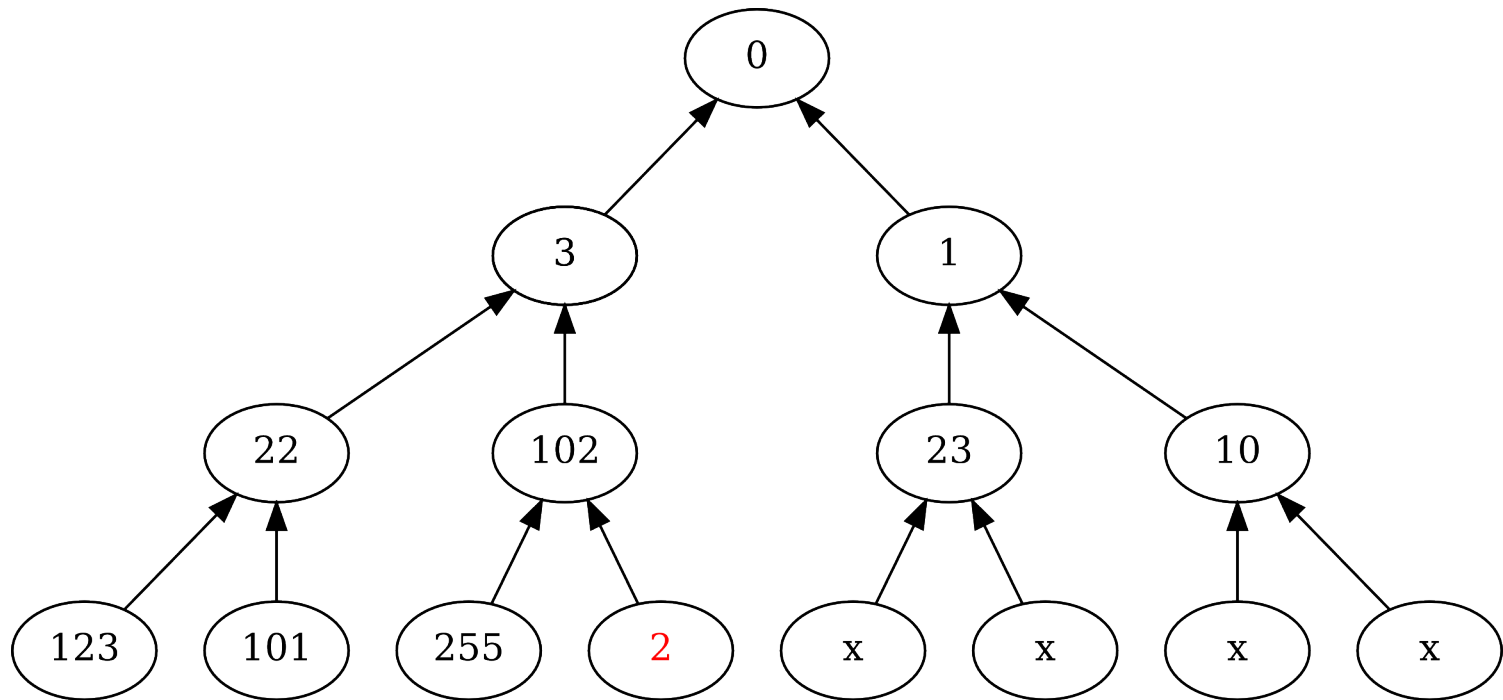
Heap (5)

- Análisis de una operación: moveUp()



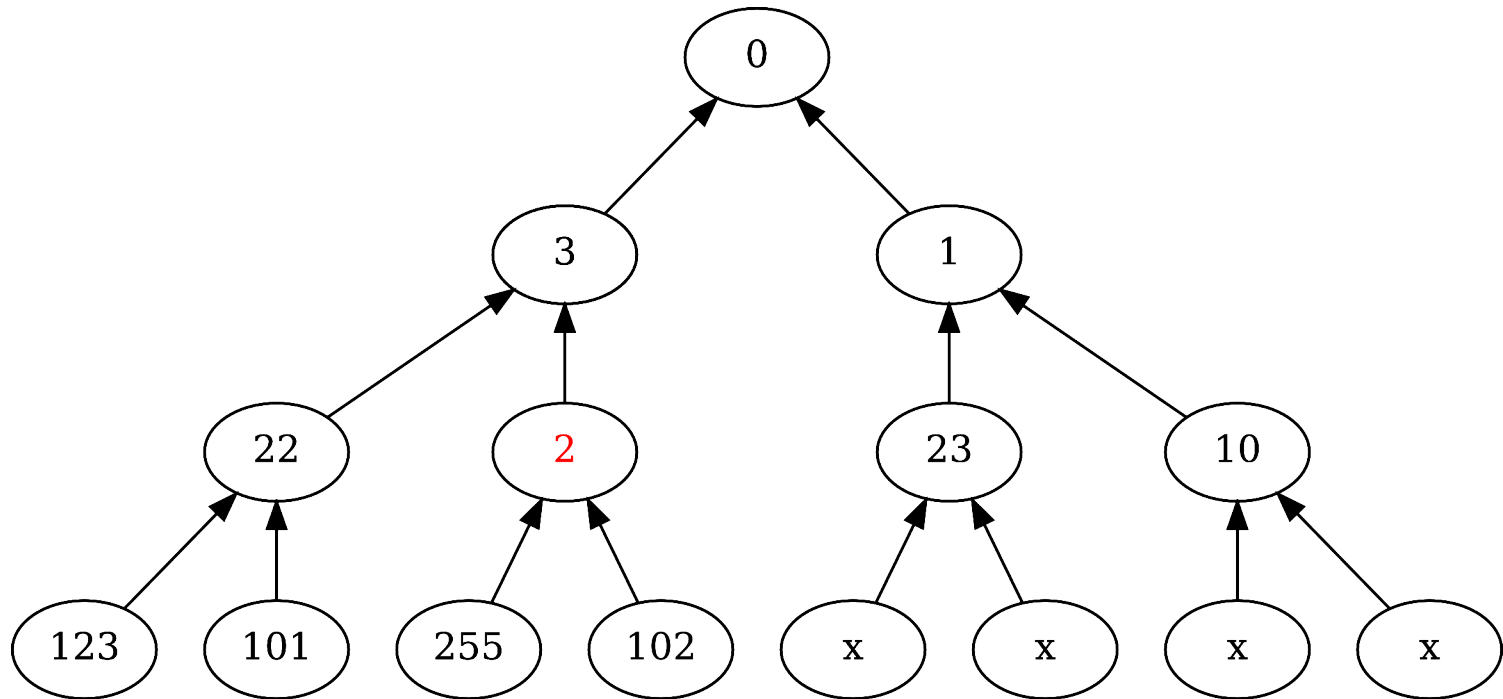
Heap (6)

- Análisis de una operación: moveUp()



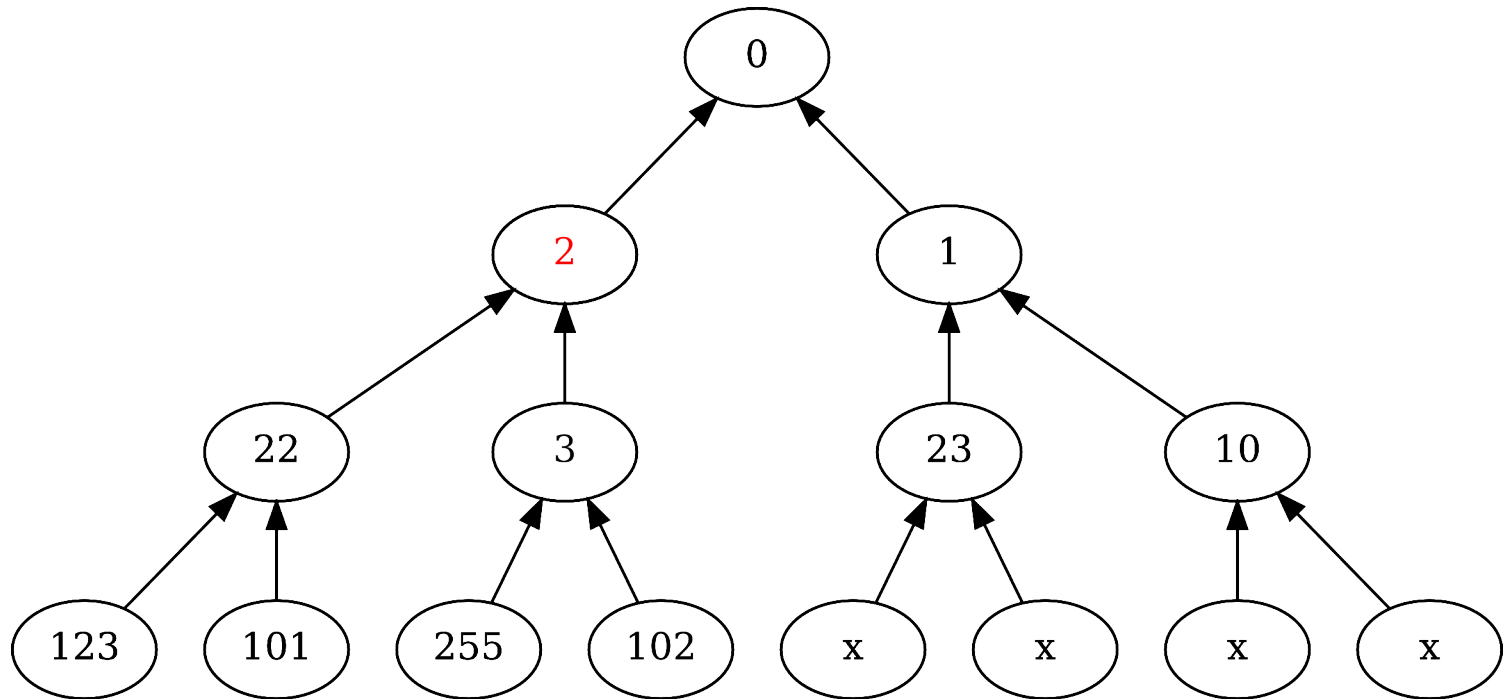
Heap (7)

- Análisis de una operación: moveUp()



Heap (8)

- Análisis de una operación: moveUp()



Heap (9)

- Implementación C

ITEM	HEAP
<pre>int label int value</pre>	<pre>int n int n_max ITEM** data</pre>

Bibliotecas

- C++: Standard Template Library
 - Completa
 - Genérica
- Boost