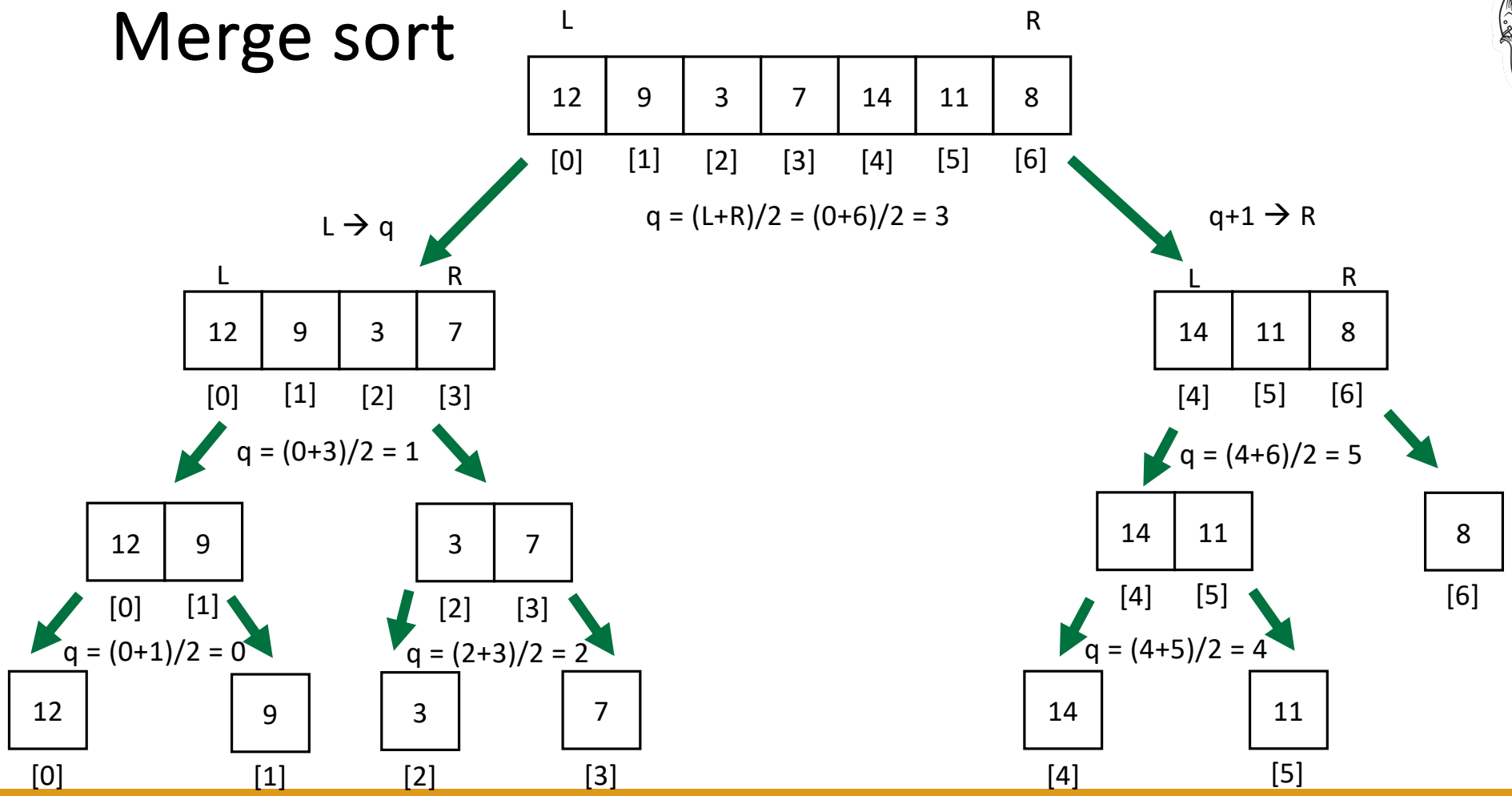
M2.3 Métodos de ordenamiento Merge Sort y Heap Sort.

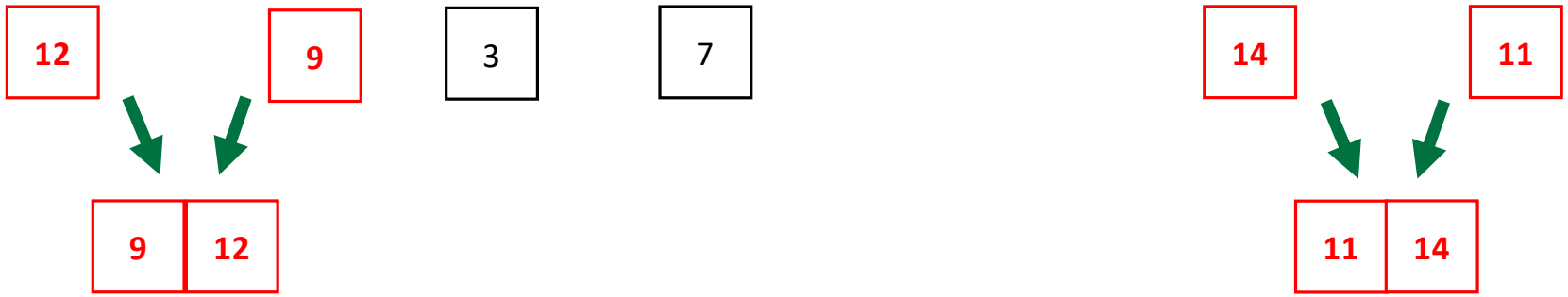
Algoritmos y Estructuras de Datos

Violeta Ocegueda

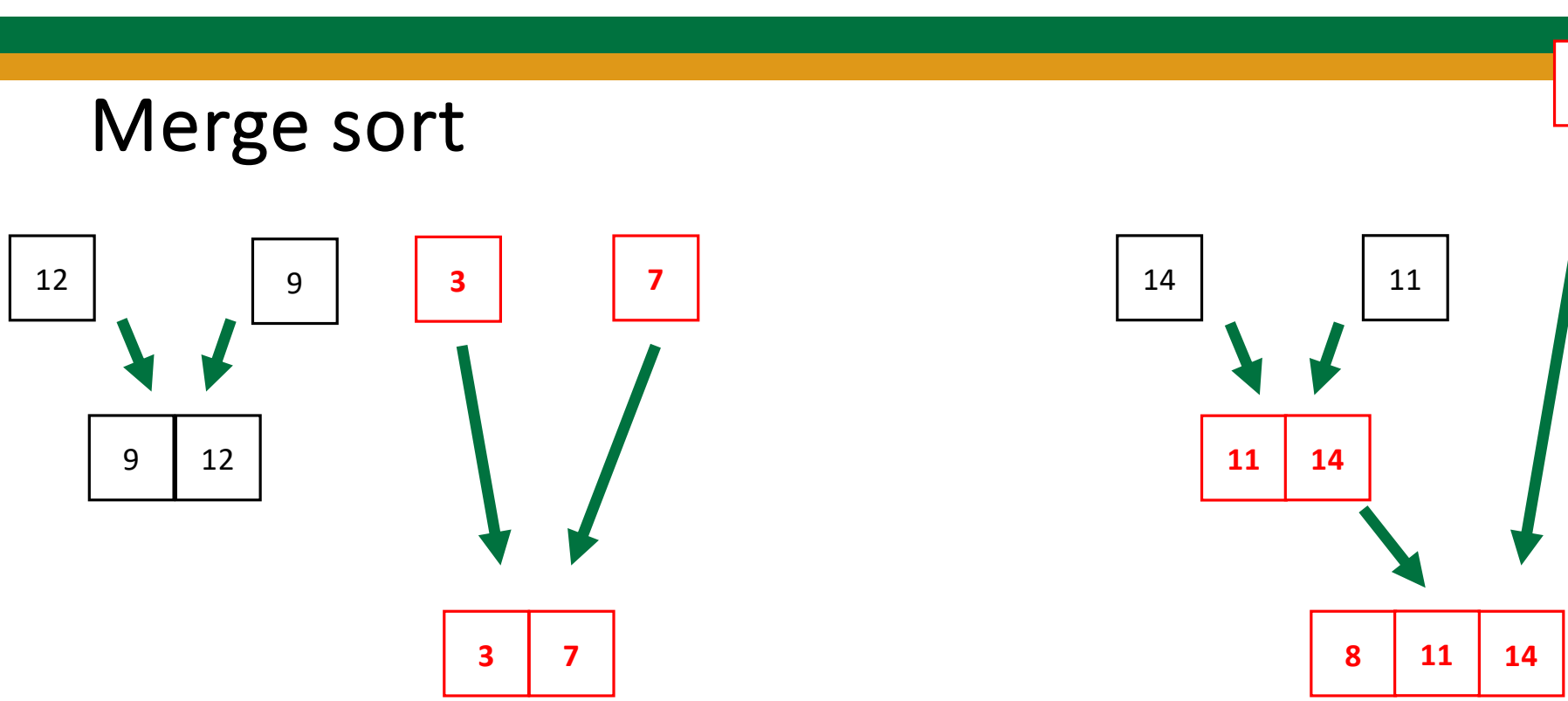
Merge sort


L
R
12 9 3 7 14 11 8
[0] [1] [2] [3] [4] [5] [6]
q = (L+R)/2 = (0+6)/2 = 3
L → q
q+1 → R
L
R
12 9 3 7
[0] [1] [2] [3]
14 11 8
[4] [5] [6]
q = (0+3)/2 = 1
q = (4+6)/2 = 5
12 9
3 7
14 11
8
[0] [1]
[2] [3]
[4] [5]
[6]
q = (0+1)/2 = 0
q = (2+3)/2 = 2
q = (4+5)/2 = 4
12
9
3
7
14
11
[0]
[1]
[2]
[3]
[4]
[5]

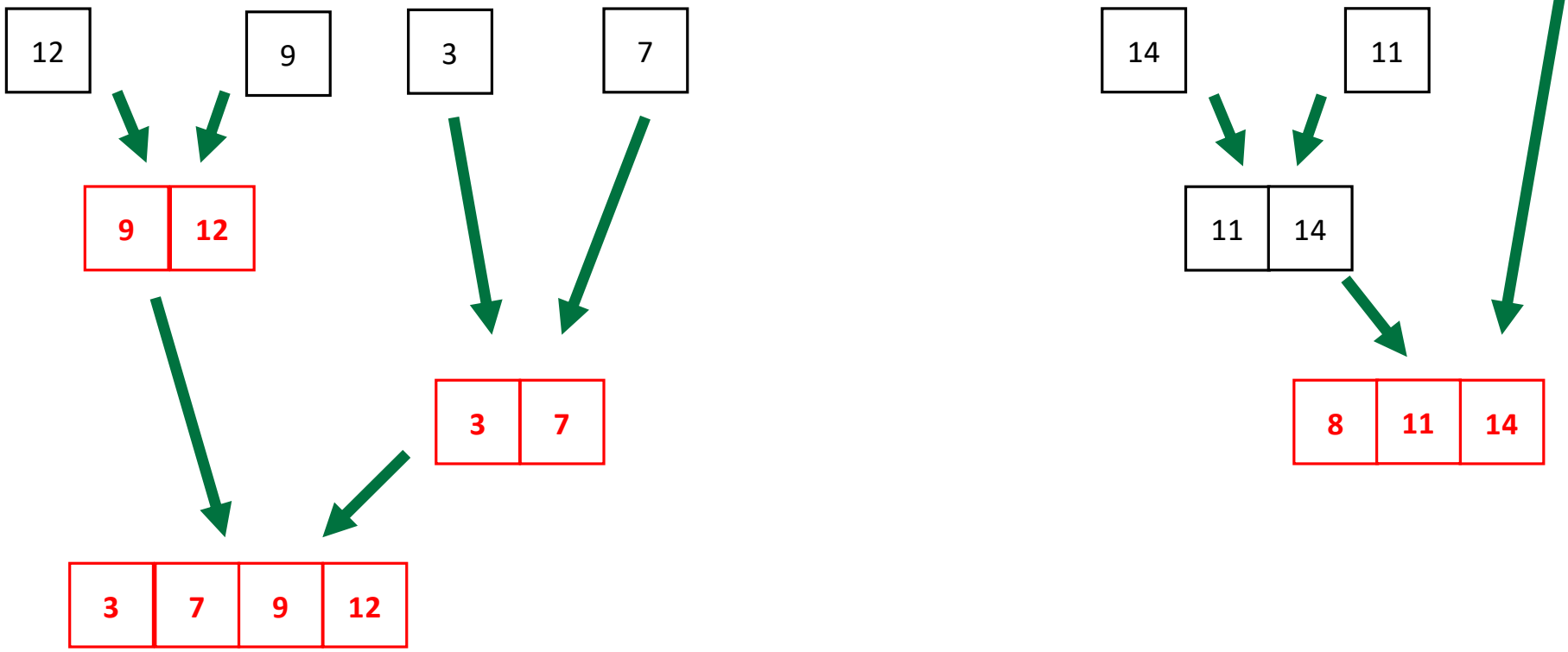
Merge sort



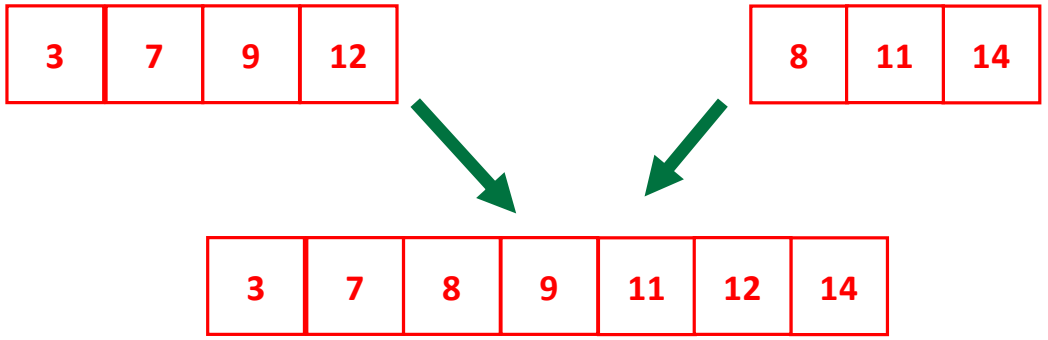
Merge sort



Merge sort



Merge sort



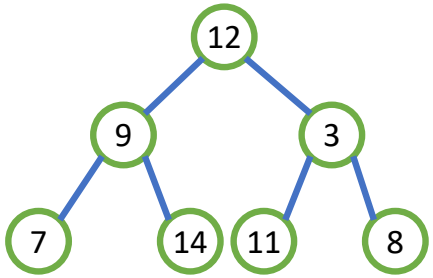
Heap sort

- Dos tipos:
  - **Max Heap**: el nodo padre tiene un valor más alto o igual que sus hijos.
  - **Min Heap**: el nodo padre tiene un valor más pequeño o igual que sus hijos.
- Consiste en:
  - Convertir el arreglo en un heap.
  - Repetir hasta que el heap tenga sólo 1 elemento:
    - Intercambiar la raíz del heap por una posición en el vector:
      - Max Heap: la raíz reemplaza la última posición disponible del vector.
      - Min Heap: la raíz reemplaza la primera posición disponible del vector.
    - Remover el último elemento del heap.
    - Heapify los elementos restantes en el heap.

Heap sort

12	9	3	7	14	11	8
[0]	[1]	[2]	[3]	[4]	[5]	[6]

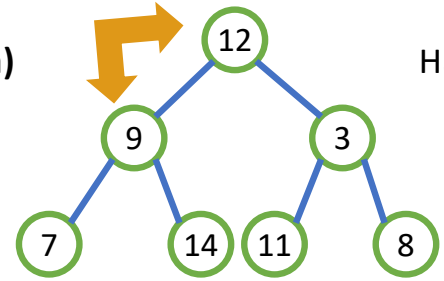
1. Construir árbol binario



Heap sort

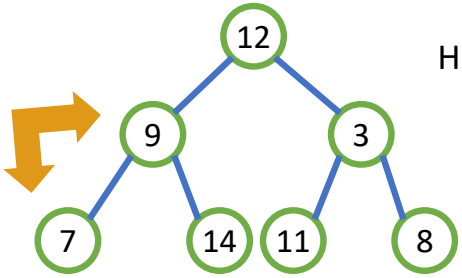
2. Convertir árbol resultante en un Max Heap (heapify)

a)



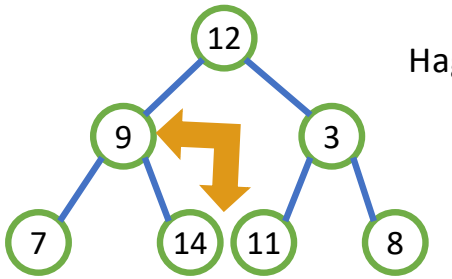
Hago intercambio? ❌

b)

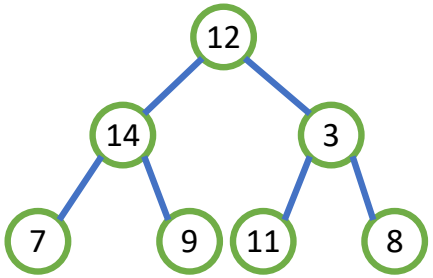


Hago intercambio? ❌

c)

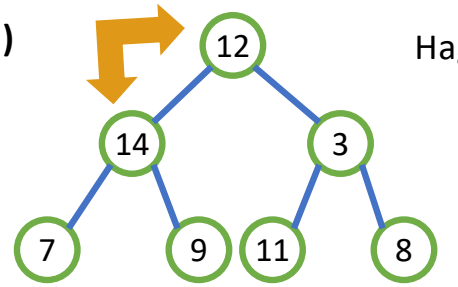


Hago intercambio? ✅

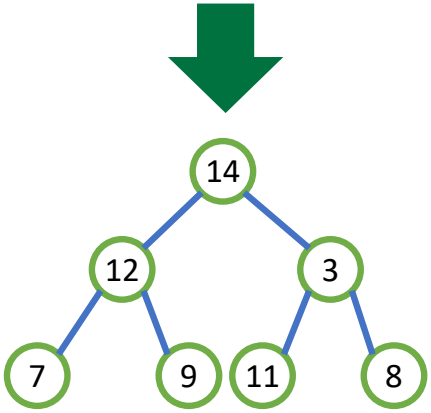


Heap sort

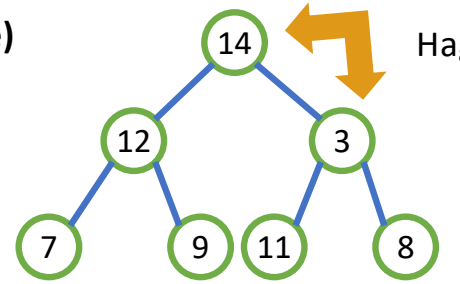
d)



Hago intercambio? ✅



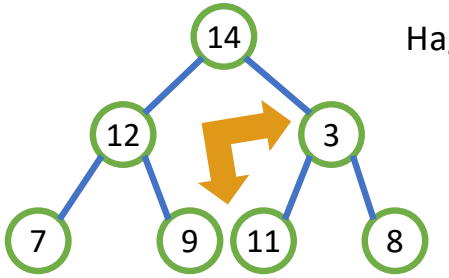
e)



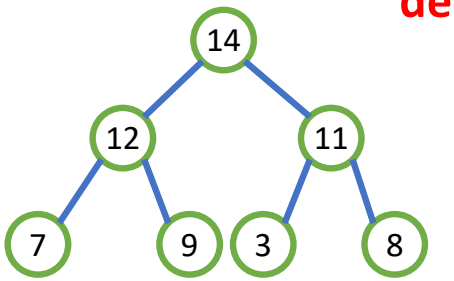
Hago intercambio? ❌

Heap sort

f)

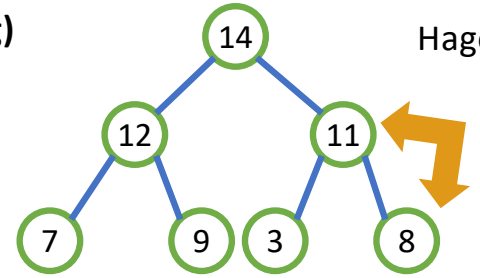


Hago intercambio? ✅



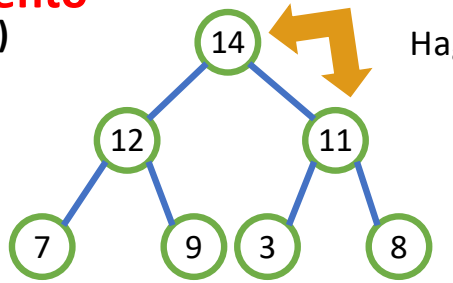
En este momento ya sé que la raíz de mi heap es el elemento de mayor valor.

g)



Hago intercambio? ❌

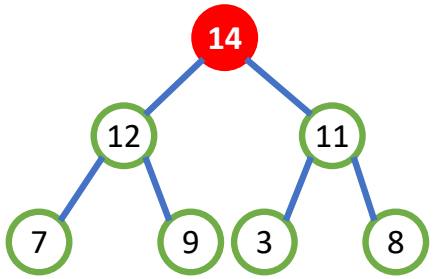
h)

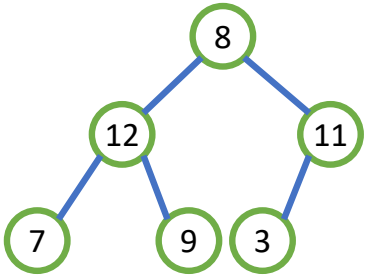


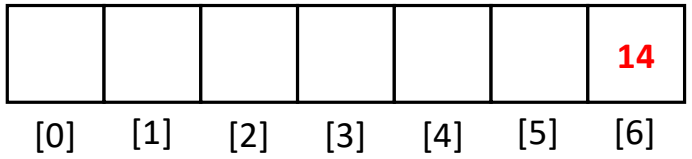
Hago intercambio? ❌

Heap sort

3. Remover la raíz de heap y sustituirlo por el último elemento del heap.



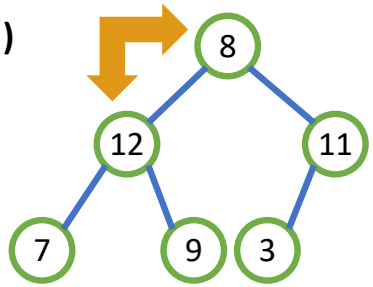




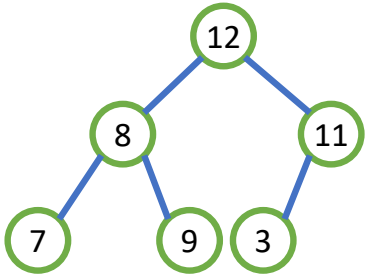
Shell sort

2. Convertir árbol resultante en un Max Heap (heapify)

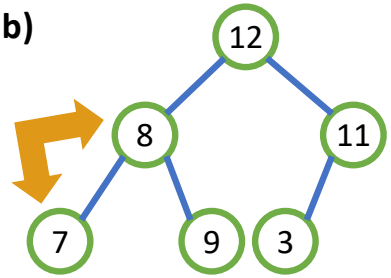
a)



Hago intercambio? ✅



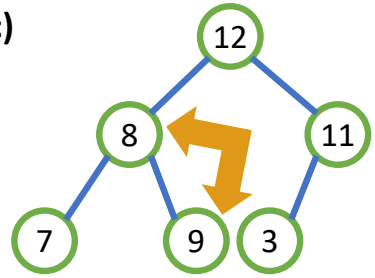
b)



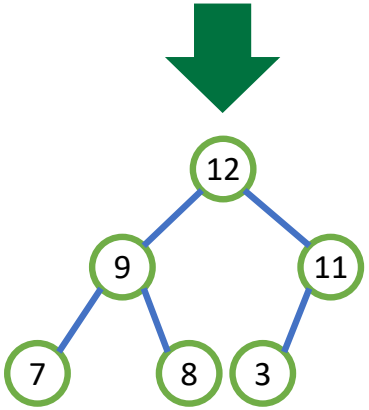
Hago intercambio? ❌

Heap sort

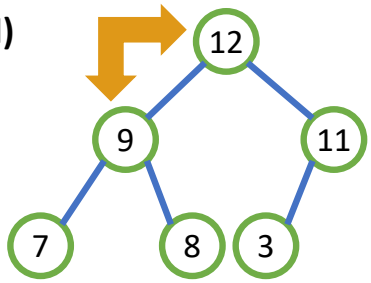
c)



Hago intercambio? ✅

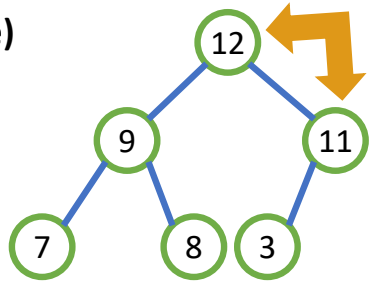


d)



Hago intercambio? ❌

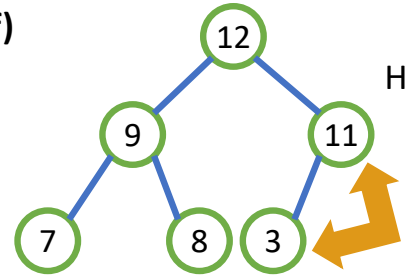
e)



Hago intercambio? ❌

Heap sort

f)

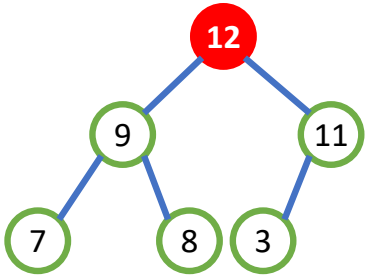


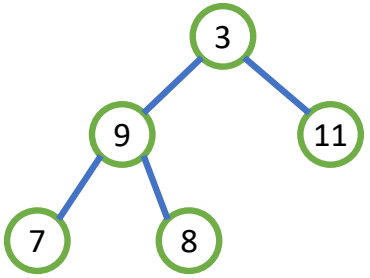
Hago intercambio? ✗

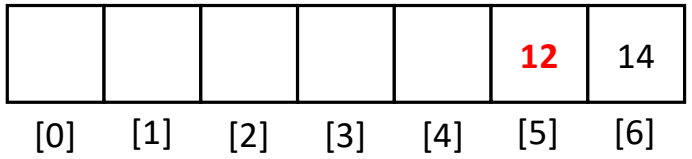
En este momento ya sé que la raíz de mi heap es el elemento de mayor valor.

Heap sort

3. Remover la raíz de heap y sustituirlo por el último elemento del heap.



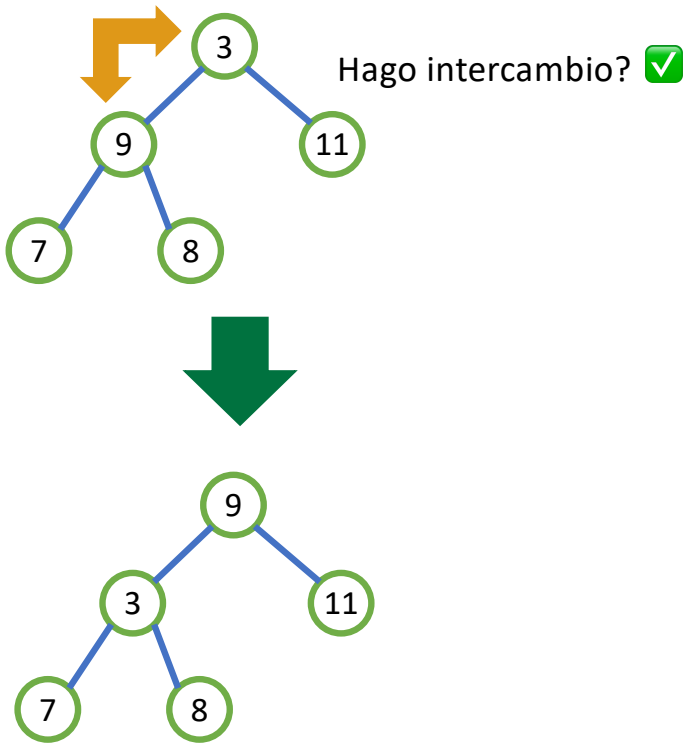




Heap sort

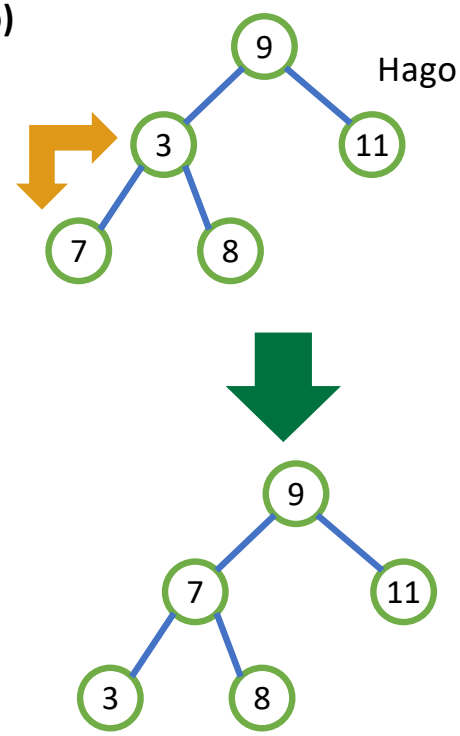
2. Convertir árbol resultante en un Max Heap (heapify)

a)



Hago intercambio? ✅

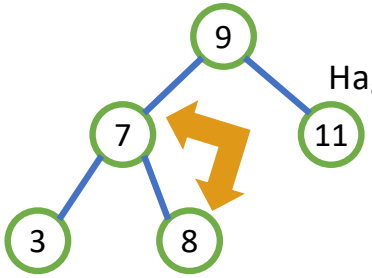
b)



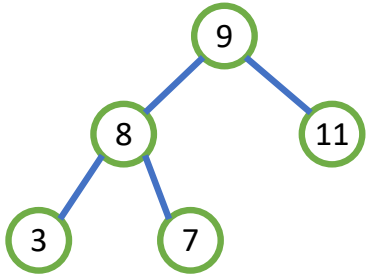
Hago intercambio? ✅

Heap sort

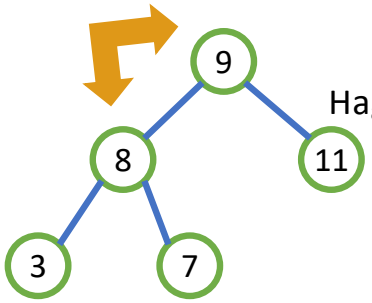
c)



Hago intercambio? ✅



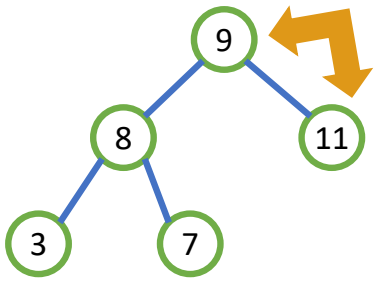
d)



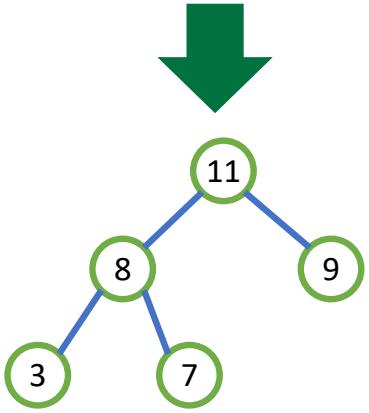
Hago intercambio? ❌

Heap sort

e)



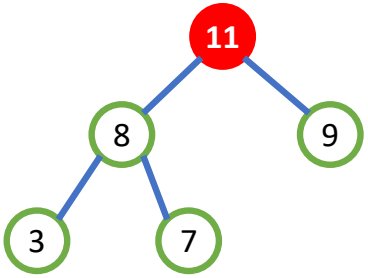
Hago intercambio? ✅

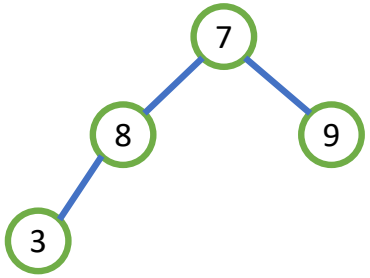


En este momento ya sé que la raíz de mi heap es el elemento de mayor valor.

Heap sort

3. Remover la raíz de heap y sustituirlo por el último elemento del heap.



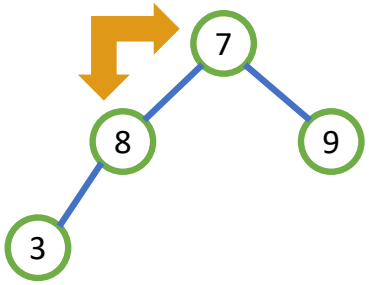


				11	12	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]

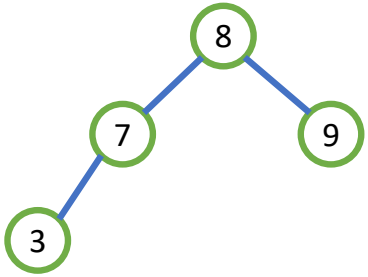
Heap sort

2. Convertir árbol resultante en un Max Heap (heapify)

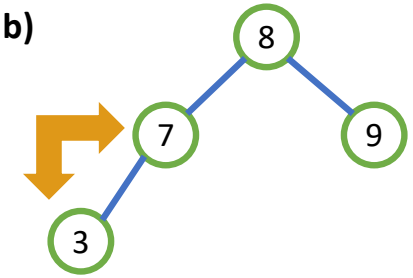
a)



Hago intercambio? ✅



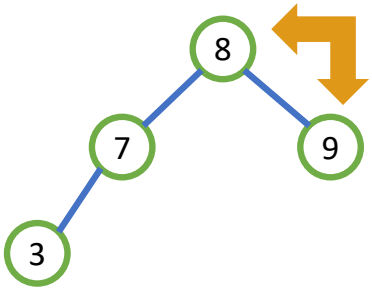
b)



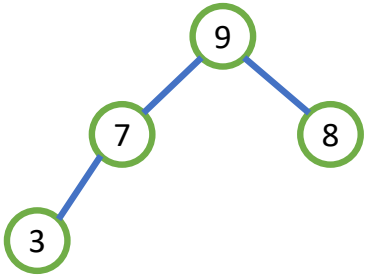
Hago intercambio? ❌

Heap sort

c)



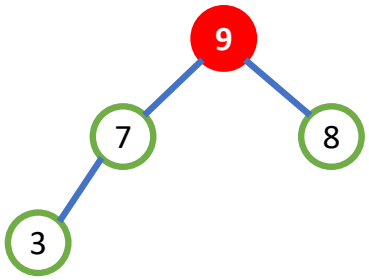
Hago intercambio? ✅

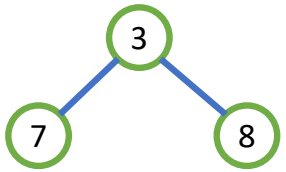


En este momento ya sé que la raíz de mi heap es el elemento de mayor valor.

Heap sort

3. Remover la raíz de heap y sustituirlo por el último elemento del heap.



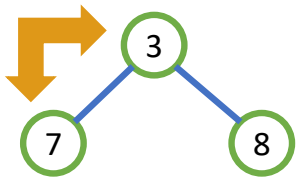


			9	11	12	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]

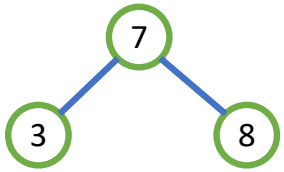
Heap sort

2. Convertir árbol resultante en un Max Heap (heapify)

a)

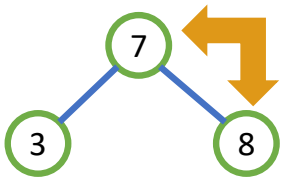


Hago intercambio? ✅

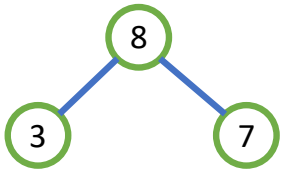


En este momento ya sé que la raíz de mi heap es el elemento de mayor valor.

b)

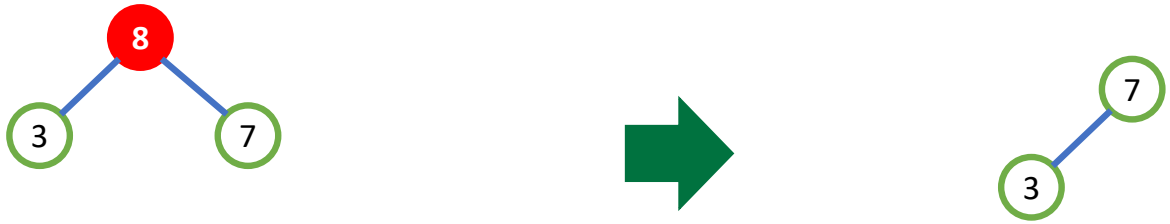


Hago intercambio? ✅



Heap sort

3. Remover la raíz de heap y sustituirlo por el último elemento del heap.

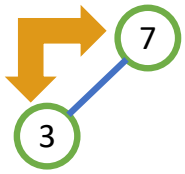


		8	9	11	12	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]

Heap sort

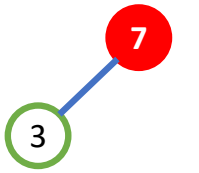
2. Convertir árbol resultante en un Max Heap (heapify)

a)



Hago intercambio? ✗

3. Remover la raíz de heap y sustituirlo por el último elemento del heap.



En este momento ya sé que la raíz de mi heap es el elemento de mayor valor.

	7	8	9	11	12	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]

Heap sort

2. Convertir árbol resultante en un Max Heap (heapify)

Pero como ya sólo queda un elemento en el heap, ese elemento va en la última posición disponible del vector.

3	7	8	9	11	12	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]