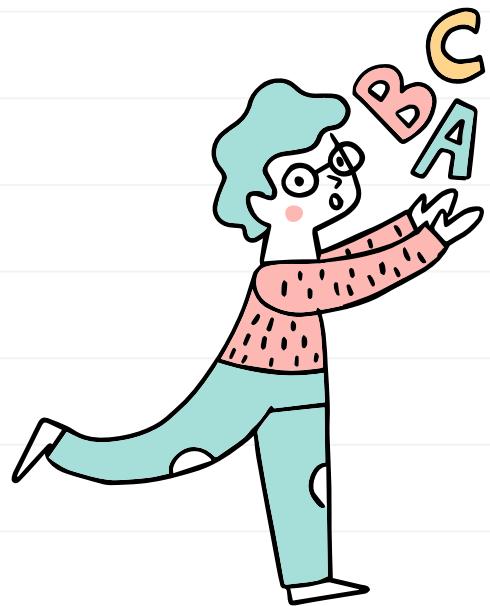


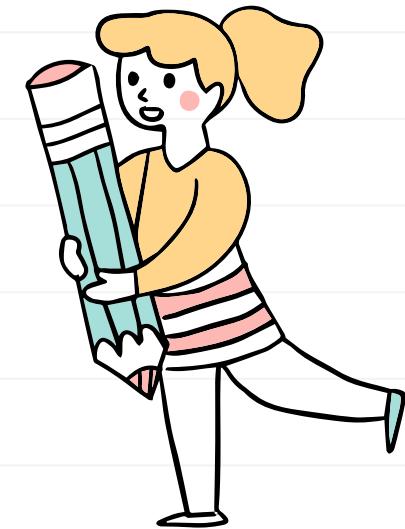
Heaps

(Montículos)

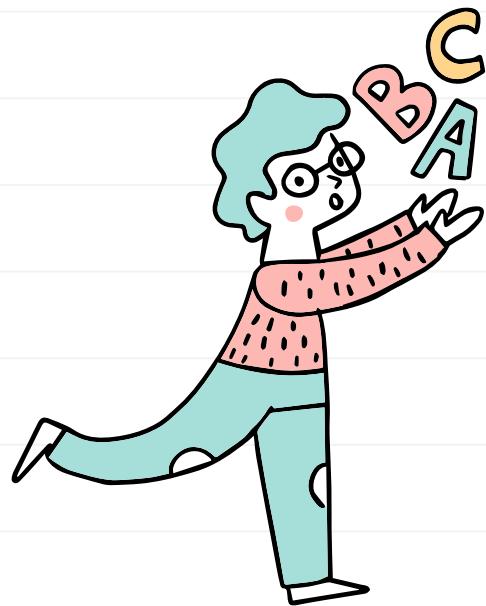
HEAPS?



Estructura de datos basada en árboles que
respetan la
“propiedad heap”

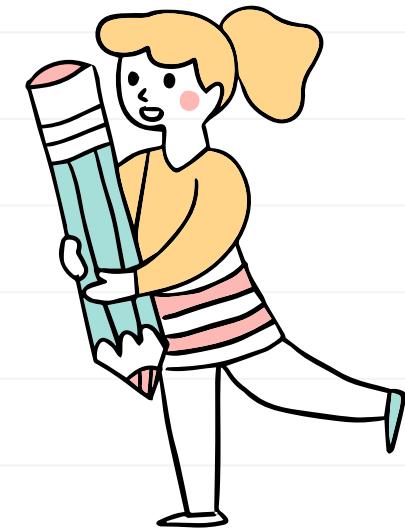


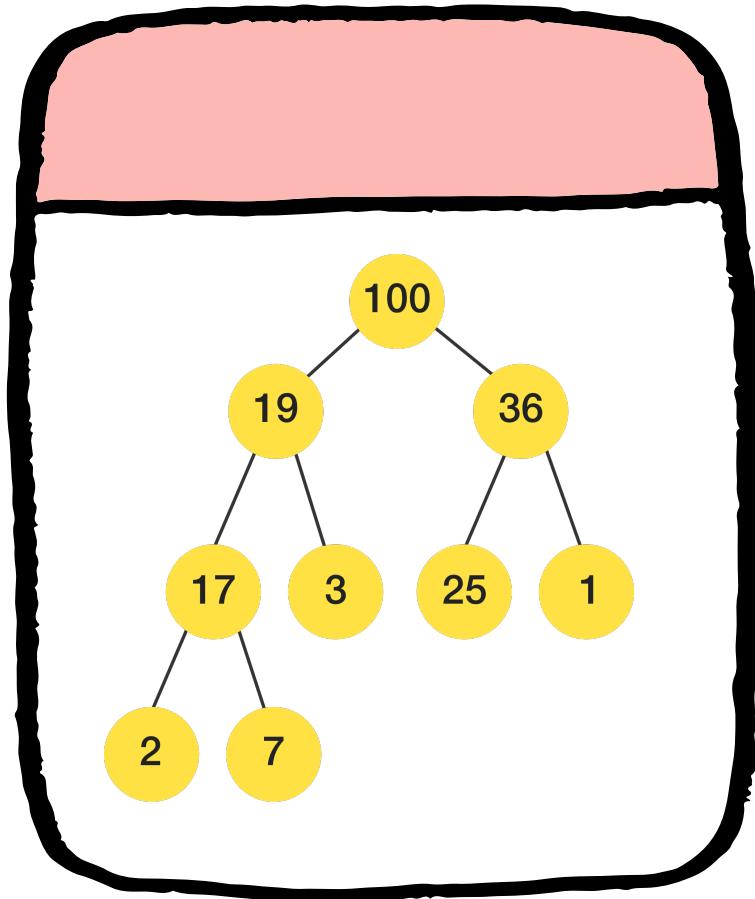
HEAPS



Estructura de datos que permite obtener el elemento mínimo (o el máximo) en un tiempo constante.

$O(1)$





Aplicaciones

Dijkstra

Algoritmo de *dijkstra* para encontrar la ruta más corta

HEAP SORT

Método de ordenamiento

Priority queues

Implementación de colas de prioridad

planificadores

De eventos, recursos, etc.

S.E.D.

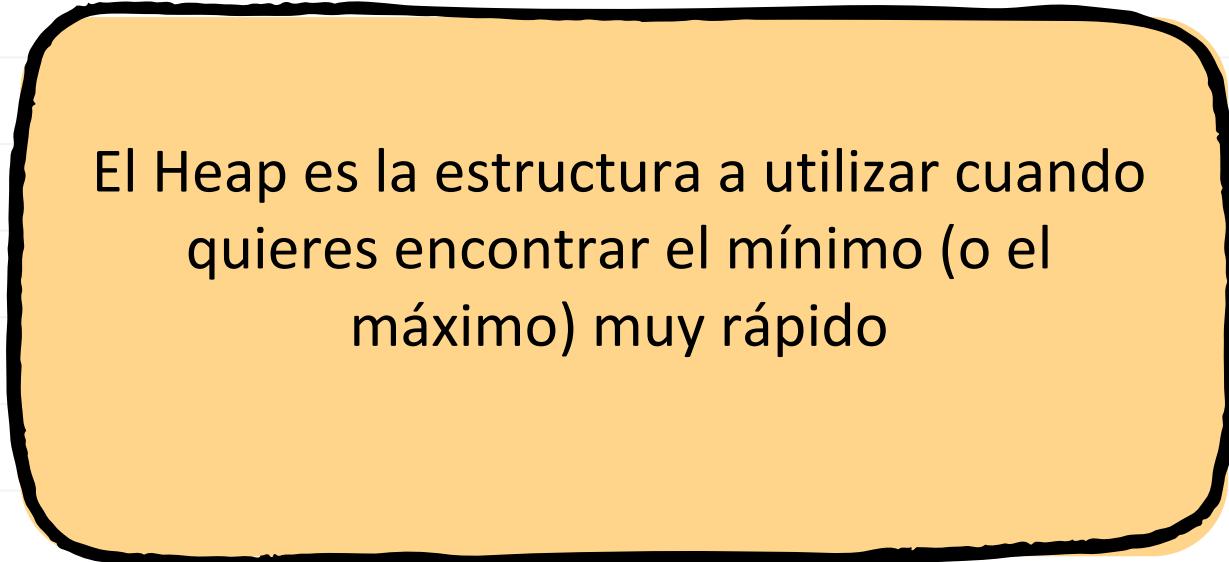
Sistemas de eventos discretos

barridos

En aplicaciones de geometría computacional



El Heap es la estructura a utilizar cuando quieras encontrar el mínimo (o el máximo) muy rápido



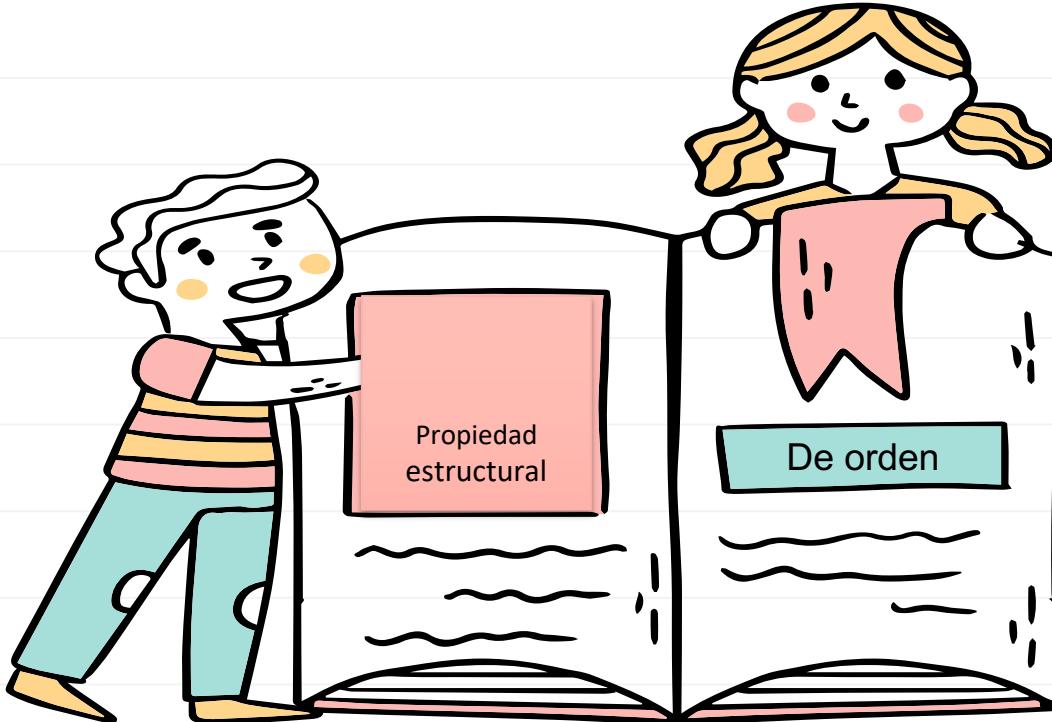
Heaps



Tipos de heaps

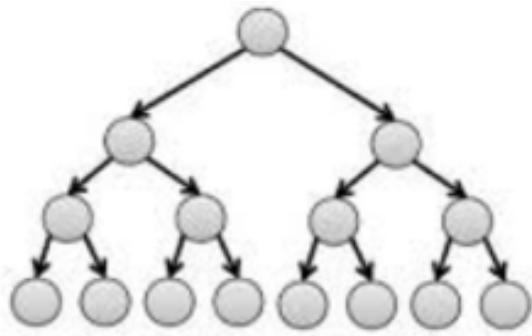
- 01
Pairing Heap
- 02
Binary Heap
- 03
Fibonacci Heap
- 04
Binomial Heap
- 05
Skew Heaps

Propiedades de un heap

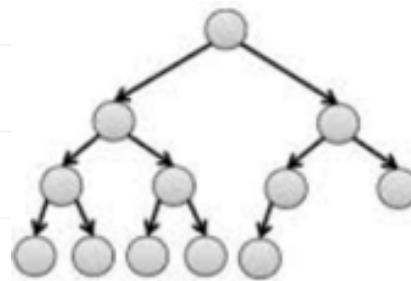


Propiedad estructural

Es un árbol binario completo



Todos los niveles están llenos, la única excepción posible es el último nivel, los nodos se deben llenar de izquierda a derecha

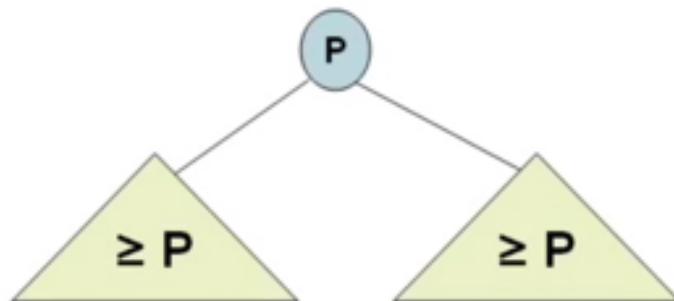


$O(\log N)$

Propiedad de orden (heap property)

Cada nodo X con padre P se cumple que el dato de P es siempre **menor o igual** que el dato en X.

El valor del padre nunca es mayor que el valor de los hijos (en un min-heap)

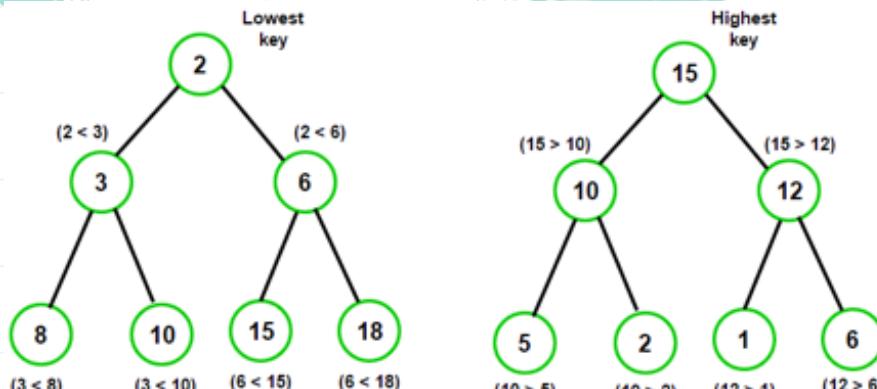


Caso especial: El nodo raíz no tiene padre

Tipos (de acuerdo a propiEdad de orden)

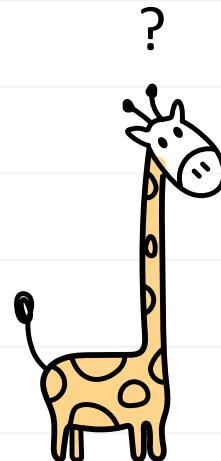


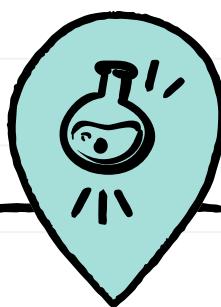
La propiedad de orden puede ser para un min-heap o max-heap



Min Heap
(Parent key is less than or equal to (\leq) the child key)

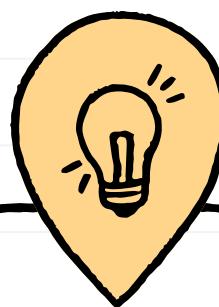
Max Heap
(Parent key is greater than or equal to (\geq) the child key)





Propiedad estructural

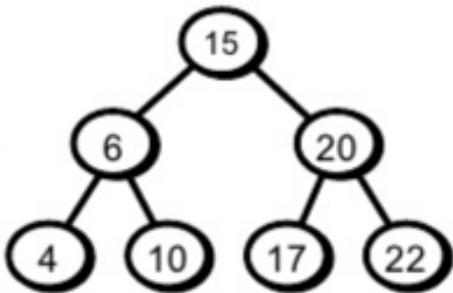
Es un árbol binario completo



Propiedad de orden

El valor del padre es mayor que sus hijos

Es un heap...?



¿Propiedad de orden?

Cada nodo mayor que su
padre



¿Propiedad estructural?

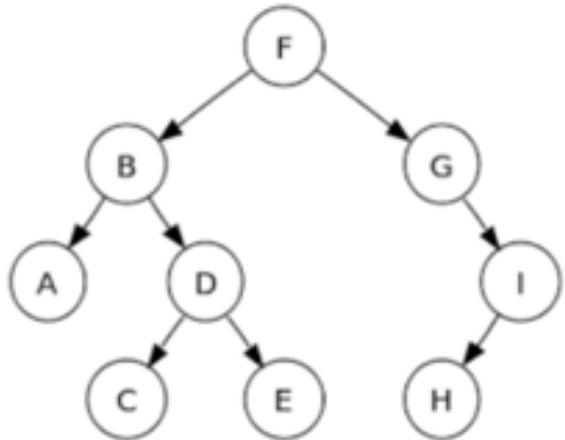
Es un arbol binario completo



¿es un heap?

Cumple las dos anteriores

Es un heap...?



¿Propiedad de orden?

Cada nodo mayor que su
padre



¿Propiedad estructural?

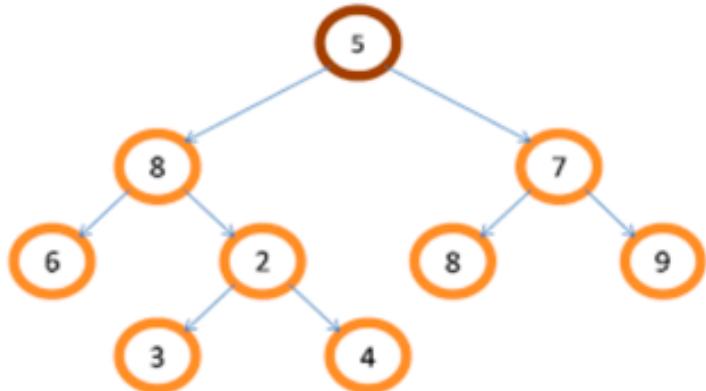
Es un arbol binario completo



¿es un heap?

Cumple las dos anteriores

Es un heap...?



¿Propiedad de orden?

Cada nodo mayor que su
padre



¿Propiedad estructural?

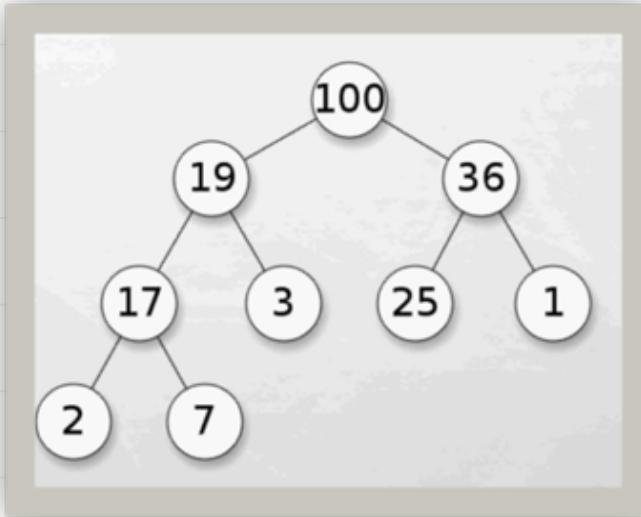
Es un arbol binario completo



¿es un heap?

Cumple las dos anteriores

Es un heap...?



¿Propiedad de orden?

Cada nodo mayor que su
padre



¿Propiedad estructural?

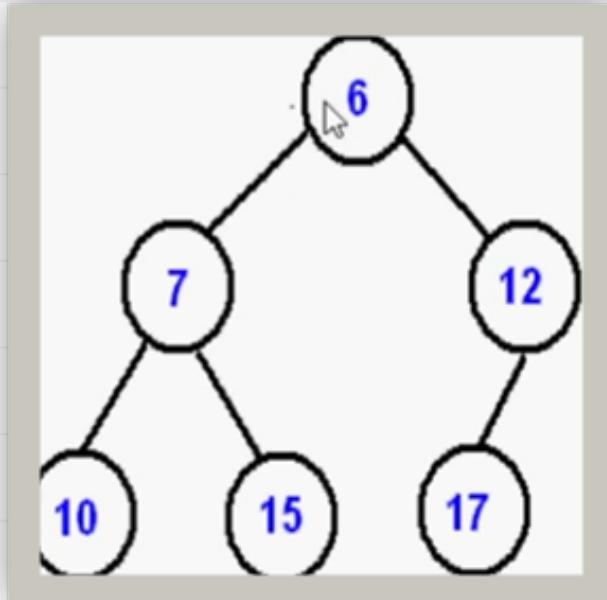
Es un arbol binario completo



¿es un heap?

Cumple las dos anteriores

Es un heap...?



¿Propiedad de orden?

Cada nodo mayor que su
padre



¿Propiedad estructural?

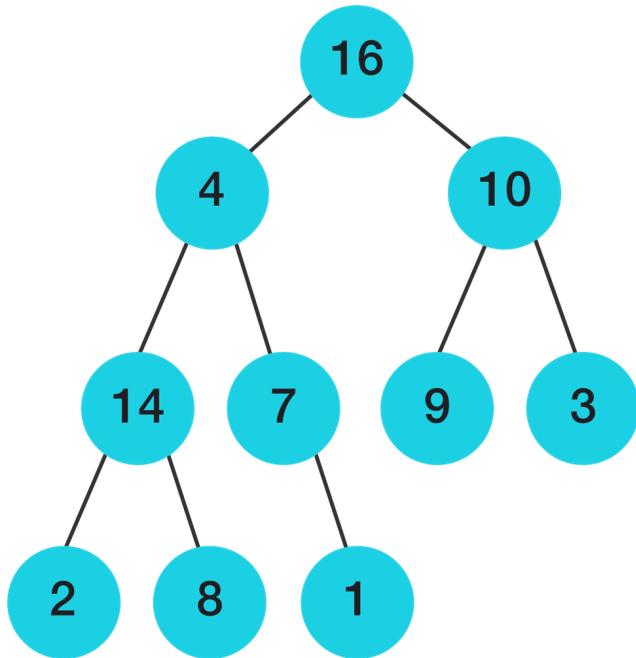
Es un arbol binario completo



¿es un heap?

Cumple las dos anteriores

Suponga que es un max_heap
Cual(es) nodos violan la propiedad heap...?



Operaciones



insertar

Agregar un elemento al
Heap

-respetando propiedad estructural
y propiedad de orden



Obtener_min

Conocer cual es el
valor mínimo
almacenado
(o el máximo si es un max-heap)



eliminar

Eliminar el valor
mínimo

No hay eliminación ni búsqueda arbitraria



John w. williams
(1929-2012)

Comparativa de costos

Type of Heap	Insert	Delete	Decrease Key	Merge	Extract Min
Binary Heap	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(n)$	$\Theta(\log n)$
Fibonacci Heap*	$O(1)$	$O(\log n)$	$O(1)$	$O(1)$	$O(\log n)$
Binomial Heap	$O(1)$	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$
Pairing Heap*	$O(1)$	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$

insertar

01

Insertar en la siguiente
posición libre

Para respetar propiedad estructural



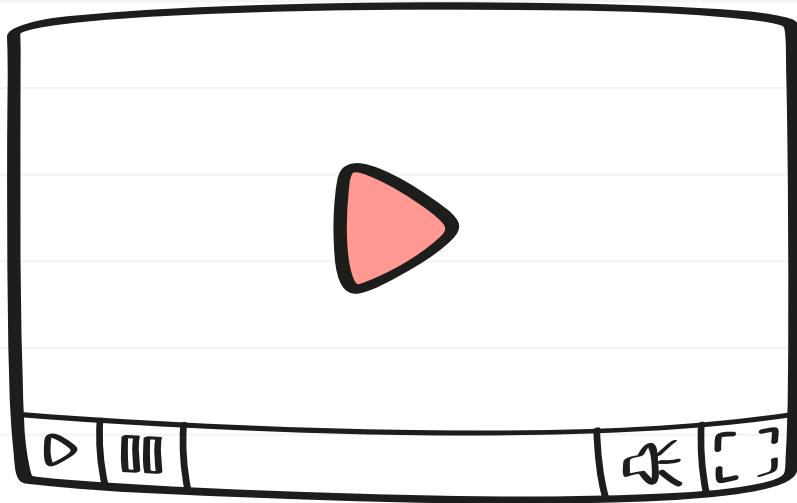
02

"Reflotar" el nuevo
elemento

Para respetar propiedad de orden

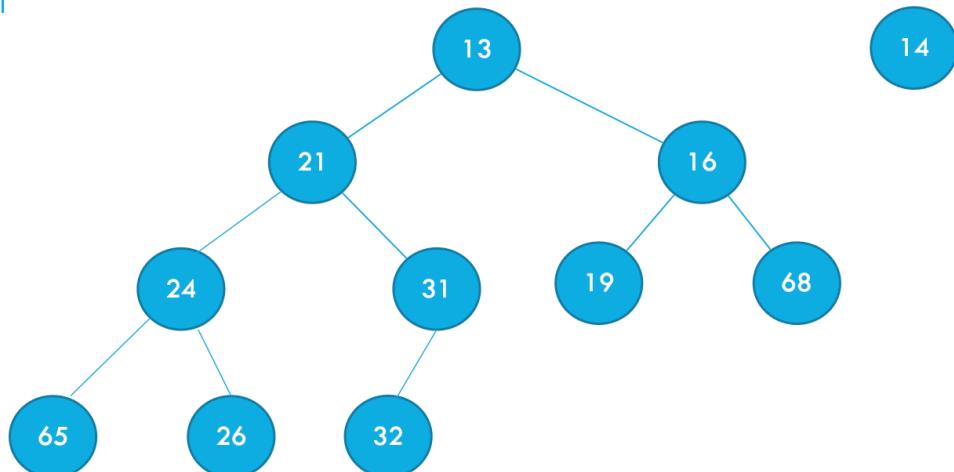
Heapify

Gayle Laakman
Cracking the Coding Interview



<https://www.youtube.com/watch?v=t0Cq6tVNRBA>

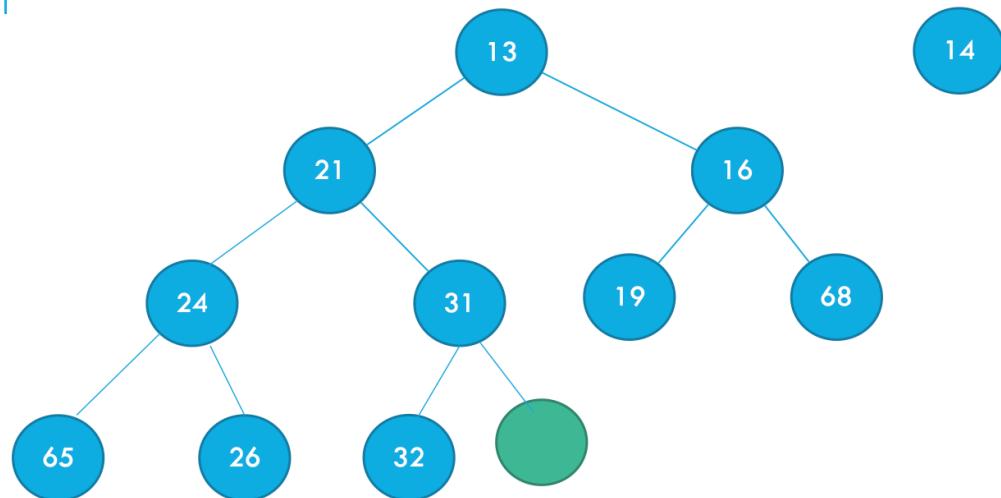
Insert(14)



El proceso de insertar requiere:

- 1) Insertar el elemento en la primer ubicación libre de acuerdo a la propiedad estructural

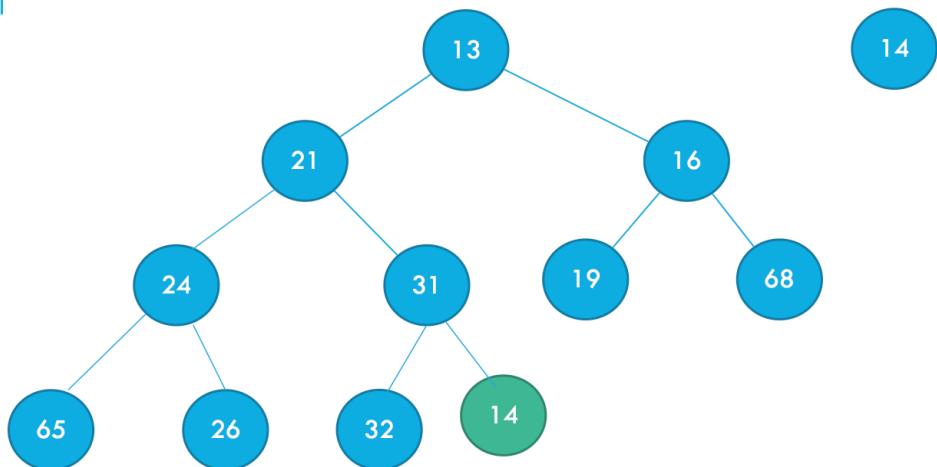
Insert(14)



El proceso de insertar requiere:

- 1) Insertar el elemento en la primer ubicación libre de acuerdo a la propiedad estructural

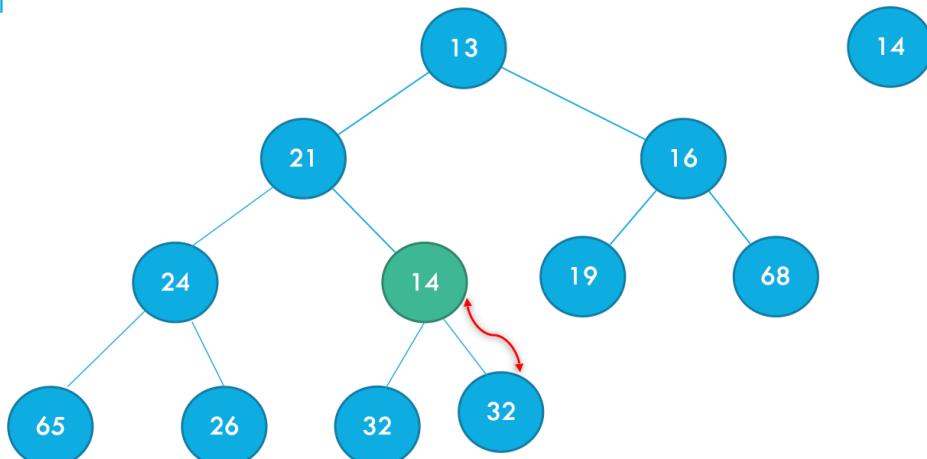
Insert(14)



El proceso de insertar requiere:

- 1) Insertar el elemento en la primer ubicación libre de acuerdo a la propiedad estructural

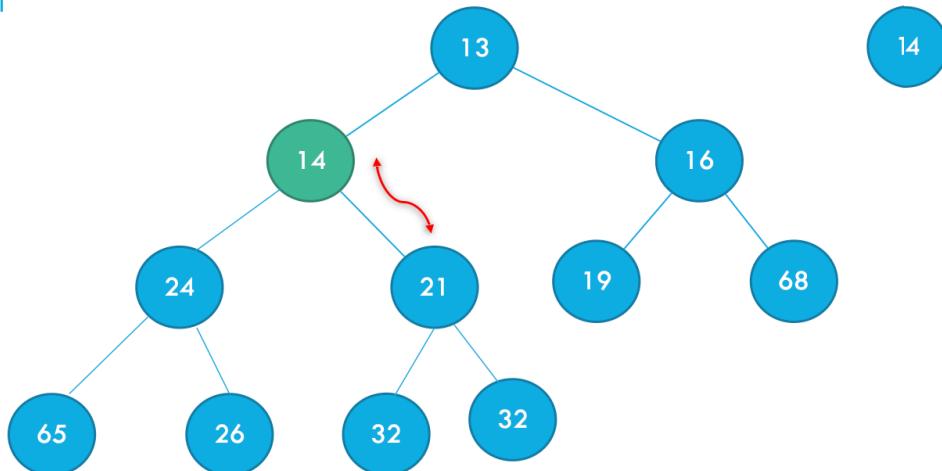
Insert(14)



El proceso de insertar requiere:

- 2) **Reflotar** el elemento de acuerdo a la propiedad de **orden**

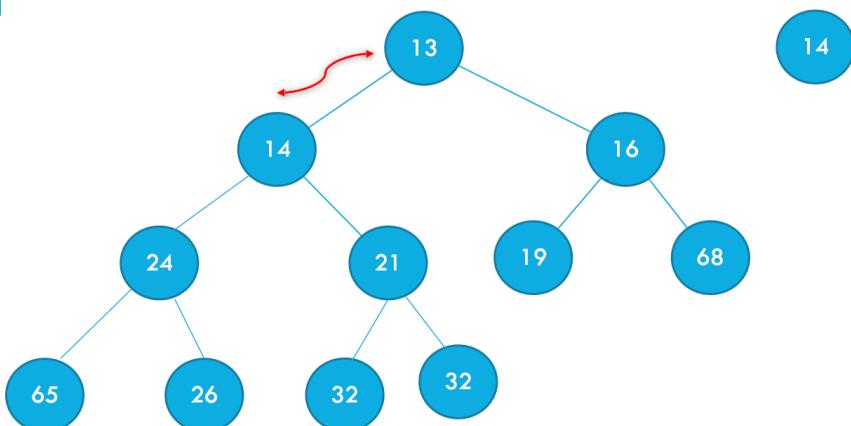
Insert(14)



El proceso de insertar requiere:

- 2) **Reflotar** el elemento de acuerdo a la propiedad de **orden**

Insert(14)



El proceso de insertar requiere:

- 2) **Reflotar** el elemento de acuerdo a la propiedad de **orden**

Costo



Mejor de
los casos

$$\Omega(1)$$



peor de
los casos

$$O(\log_2(N))$$

eliminar

01

Sustituir el valor de la raíz por el
último elemento

Para respetar propiedad estructural

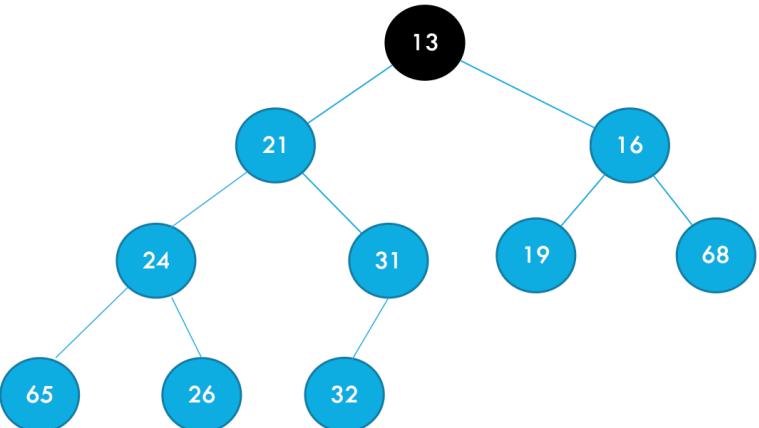


02

"Hundir" el valor en la
raíz

Para respetar propiedad de orden

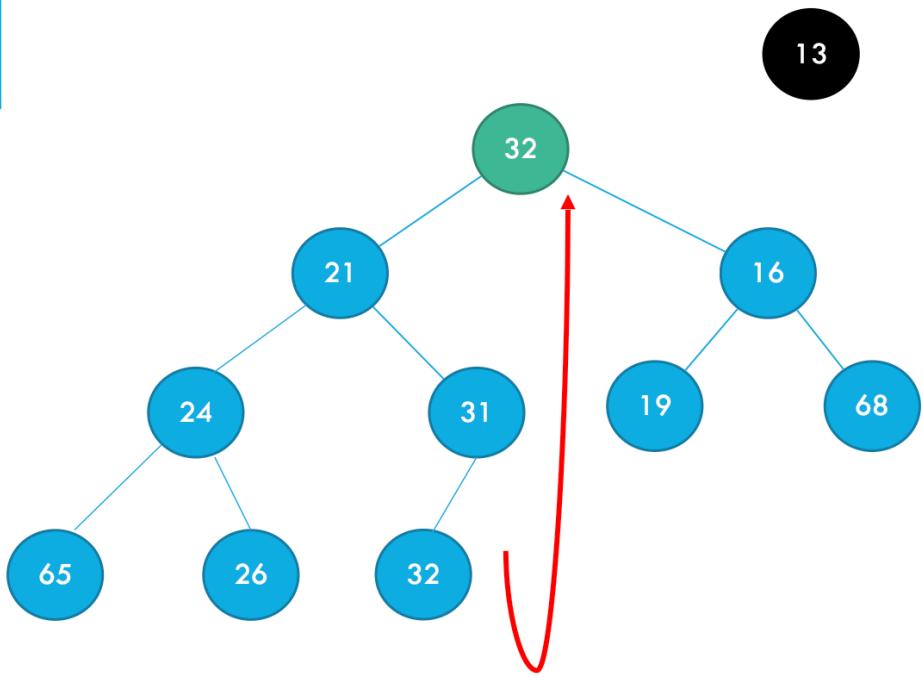
Eliminar()



El proceso de eliminar requiere:

- 1) **Sustituir** el la raíz por el ultimo elemento

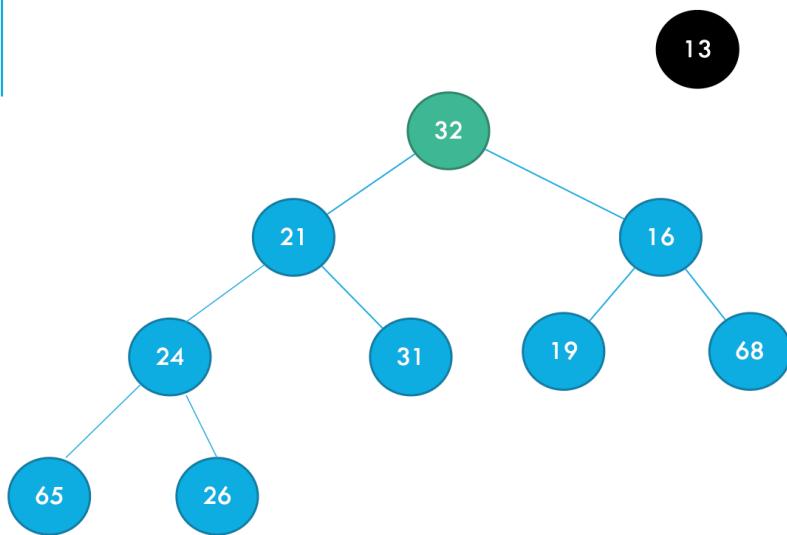
Eliminar()



El proceso de eliminar requiere:

- 1) *Sustituir* la raíz por el ultimo elemento

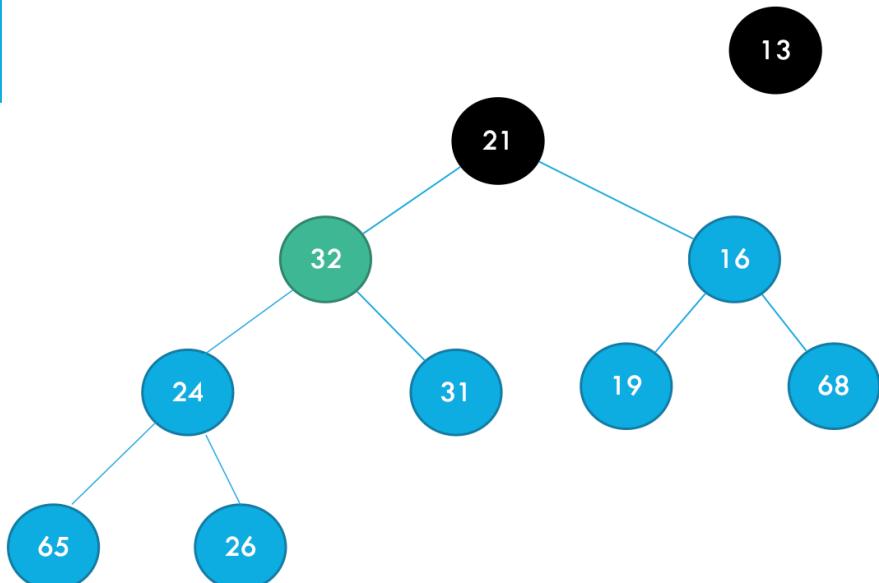
Eliminar()



El proceso de eliminar requiere:

- 2) **Hundir** el valor en la raíz

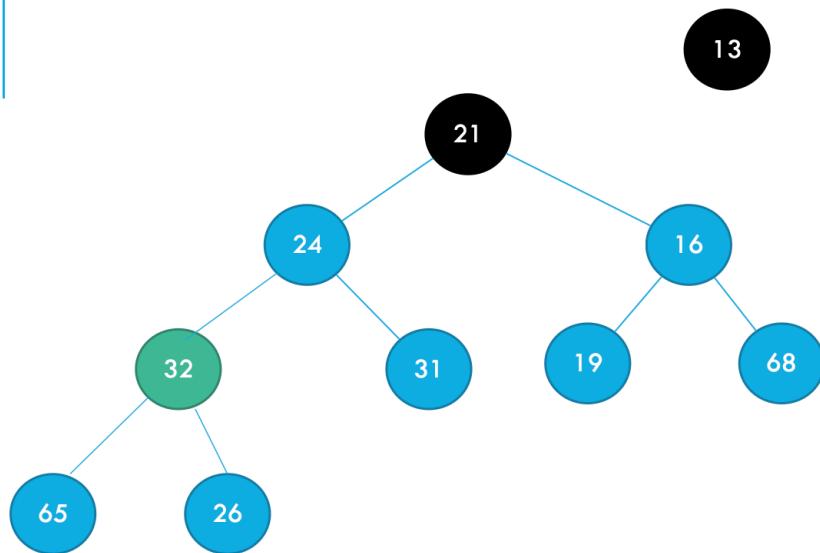
Eliminar()



El proceso de eliminar requiere:

2) **Hundir** el valor en la raíz

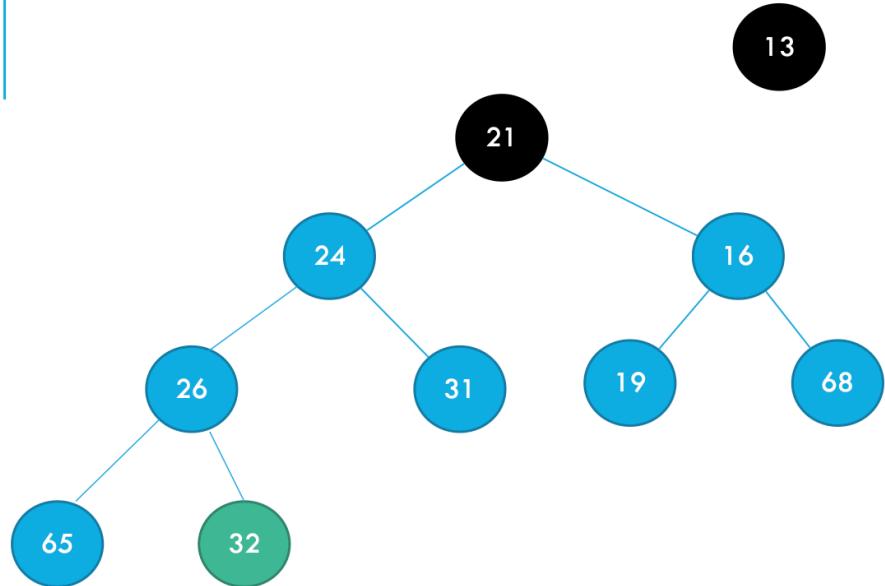
Eliminar()



El proceso de eliminar requiere:

- 2) **Hundir** el valor en la raíz

Eliminar()



El proceso de eliminar requiere:

- 2) **Hundir** el valor en la raíz

Costo



Mejor de
los casos

$$\Omega(1)$$



peor de
los casos

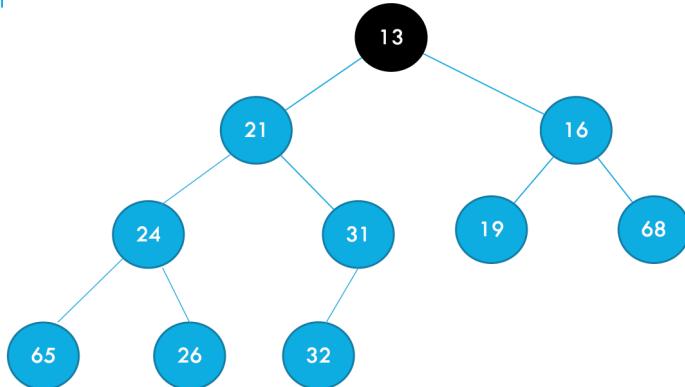
$$O(\log_2(N))$$

Obtener_min()

01

Regresar el valor en la raiz

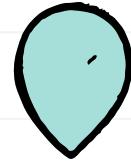
Obtener_min()



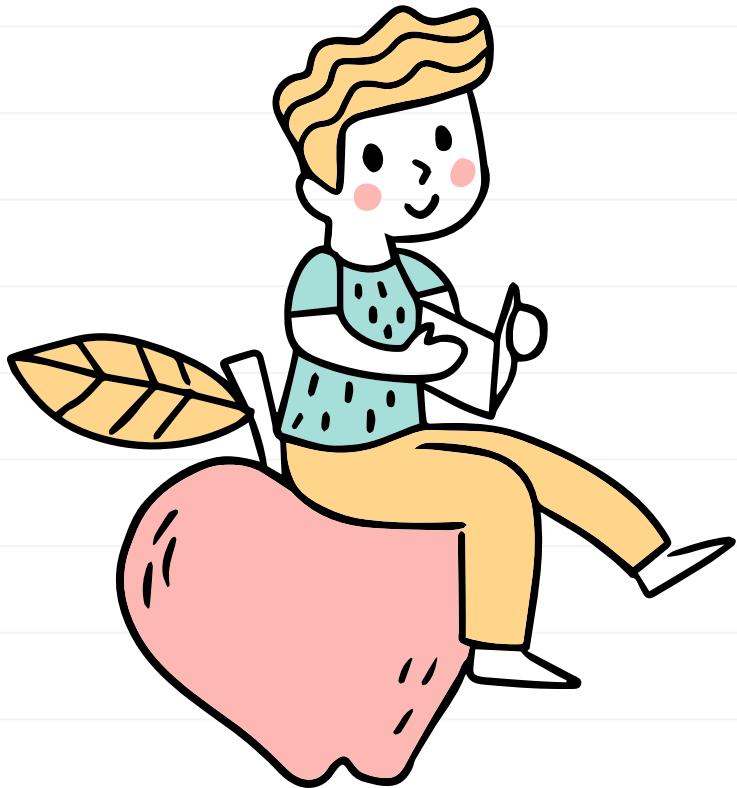
El proceso de obtener_min requiere:

- 1) **Regresar** el valor en la raíz

Costo



$O(1)$



Dudas?



CREDITS: This presentation template was created by [Slidesgo](#),
including icons by [Flaticon](#), and infographics & images by [Freepik](#).