Ejercitación 3 – Heap

Ejercicio 1

A partir de una heap inicialmente vacía, inserte de a uno los siguientes valores:

6, 4, 15, 2, 10, 11, 8, 1, 13, 7, 9, 12, 5, 3, 14

Respuesta:

[6]

[4, 6]

[4, 6, 15]

[2, 4, 15, 6]

[2, 4, 15, 6, 10]

[2, 4 ,11, 6, 10, 15]

[2, 4, 8, 6, 10, 15, 11]

[1, 2, 8, 4, 10, 15, 11, 6]

[1, 2, 8, 4, 10, 15, 11, 6, 13]

[1, 2, 8, 4, 7, 15, 11, 6 ,13, 10]

[1, 2, 8, 4, 7, 15, 11, 6, 13, 10, 9]

[1, 2, 8, 4, 7, 12, 11, 6, 13, 10, 9, 15]

[1, 2, 5, 4, 7, 8, 11, 6, 13, 10, 9, 15, 12]

[1, 2, 3, 4, 7, 8, 5, 6, 13, 10, 9, 15, 12, 11]

[1, 2, 3, 4, 7, 8, 5, 6, 13, 10, 9, 15, 12, 11, 14]

Explicación:

Salvo que el enunciado diga lo contrario, se asume que es una minHeap. En una minHeap la raiz (i = 1) es el elemento minimo. Sus hijos son iguales o mayores. Y sus posiciones en la Heap son 2*i para el hijo izquierdo y 2*i + 1 para el hijo derecho.

El proceso de armado empieza insertando el nuevo elemento al final del arreglo y haciendo un filtrado, donde se compara el nuevo elemento con su padre (segun la posicion en el arreglo se sabe cual es el padre). Si este es mayor, se intercambian los elementos, se vuelve a verificar con el nuevo padre (si hay) y en caso de ser menor al nuevo padre se vuelve a intercambiar, asi continua hasta que no sea posible.

Por ejemplo al insertar el 2:

Previamente la Heap es: [4, 6, 15] al insertar el 2 sin filtrar seria [4, 6, 15, 2] posicion 4, es decir, su padre es el 6 que esta en la posicion 2. Como 6 > 2 se intercambian y la heap quedaria [4, 2, 15, 6] su nuevo padre es la raiz del arbol, 4, como 4 > 2 se intercambian y la heap queda [2, 4, 15, 6] ahora cumple la propiedad de la minHeap y asi queda.

Ejercicio 2

a) ¿Cuántos elementos hay, al menos, en una heap de altura h?

b) ¿Dónde se encuentra ubicado el elemento mínimo en una max-heap?

c) ¿El siguiente arreglo es una max-heap : [23, 17, 14, 6, 13, 10, 1, 5, 7, 12] ?

Respuesta:

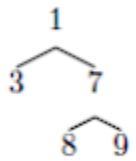
a) 2^h elementos como minimo ya que una heap es un arbol binario completo
b) En una hoja
c) No, para que sea una maxHeap la raiz debe ser mayor o igual que sus hijos, en el caso del elemento 7, su posicion en el arreglo es 9, quiere decir que es el hijo derecho del elemento 4, que es 6 y 6 no es mayor que 7.

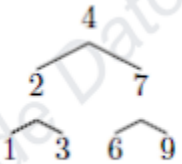
Explicación:

NA

Ejercicio 3

Dados los siguientes árboles, indique si representan una heap. Justifique su respuesta.





Respuesta:

Ninguno representa una heap.

Explicación:

El primer arbol no es un arbol binario completo ya que los nodos del ultimo nivel no estan a partir del lado izquierdo del arbol.

El segundo arbol no cumple la propiedad de minHeap ni de maxHeap, 4 es mayor que 2 pero menor que 7.

Ejercicio 4

Dibuje todas las min-heaps posibles para este conjunto de claves: {A, B, C, D, E}

Respuesta:

Explicación:

Ejercicio 5

A partir de una min-heap inicialmente vacía, dibuje la evolución del estado de la heap al ejecutar las

siguientes operaciones:

Insert(5), Insert(4), Insert(7), Insert(1), DeleteMin(), Insert(3), Insert(6), DeleteMin(), DeleteMin(), Insert(8), DeleteMin(), Insert(2), DeleteMin(), DeleteMin().

Respuesta:

insert(5): [5]

insert(4): [4, 5]

insert(7): [4, 5, 7]

insert(1): [1, 4, 7, 5]

DeleteMin(): [4, 5, 7]

insert(3): [4, 3, 7, 5]

insert(6): [4, 3, 7, 5, 6]

DeleteMin(): [3, 5, 7, 6]

DeleteMin(): [5, 6, 7]

insert(8): [5, 6, 7, 8]

DeleteMin(): [6, 8, 7]

insert(2): [2, 6, 7, 8]

DeleteMin(): [6, 8, 7]

DeleteMin(): [7, 8]

Explicación:

La operacion insert() es la explicacion del armado de la heap (ejercicio 1). DeleteMin() va a eliminar el elemento minimo, que en una minHeap siempre es la raiz en posicion 1, al eliminarlo el ultimo elemento del arreglo pasa a ser la nueva raiz. Esto puede romper la propiedad de la minHeap asi que se filtra para restaurarla.

Ejemplo: DeleteMin(): [3, 5, 7, 6]
Se elimina la raiz y el ultimo elemento pasa a ser el primero, el arreglo queda [6, 5, 7]. Al compararse con sus dos hijos, 5, el hijo izquierdo es menor que 6, por lo tanto se intercambian y la heap queda [5, 6, 7]

Ejercicio 6

Aplique el algoritmo BuildHeap para construir una minHeap en tiempo lineal con los siguientes valores: [150, 80, 40, 10, 70, 110, 30, 120, 140, 60, 50, 130, 100, 20, 90]

Respuesta:

[150, 80, 40, 10, 70, 110, 20, 120, 140, 60, 50, 130, 100, 30, 90]

[150, 80, 40, 10, 70, 100, 20, 120, 140, 60, 50, 130, 110, 30, 90]

[150, 80, 40, 10, 50, 100, 20, 120, 140, 60, 70, 130, 110, 30, 90]

[150, 80, 20, 10, 50, 100, 30, 120, 140, 60, 70, 130, 110, 40, 90]

[150, 10, 20, 80, 50, 100, 30, 120, 140, 60, 70, 130, 110, 40, 90]

[10, 50, 20, 80, 60, 100, 30, 120, 140, 150, 70, 130, 110, 40, 90]

Explicación:

Insertar los elementos de uno en uno tiene un tiempo de ejecucion de O(log n) por cada elemento. Cuando se pide que se haga en tiempo lineal, es decir, O(n) se filtra desde el ultimo padre hacia arriba. Este metodo se utiliza cuando tenemos todos los elementos y unicamente hay que ordenarlos, no es posible hacerlo si debemos insertar de uno en uno los elementos.

Para saber cual es el ultimo padre se toma el ultimo indice del arreglo y se divide entre 2, en este caso el ultimo indice es 15 y entre 2 - 1 es 7, el ultimo padre es el elemento 30, se compara con sus hijos 20 y 90 y se intercambia con 20 para cumplir la propiedad de minHeap. Esto se realiza con cada padre hasta finalmente ordenar todo el arreglo como un minHeap.

Ejercicio 7

Aplique el algoritmo HeapSort, para ordenar descendentemente los siguientes elementos:

{15, 18, 40, 1, 7, 10, 33, 2, 140, 500, 11, 12, 13, 90}

Muestre paso a paso la ejecución del algoritmo sobre los datos.

Respuesta:

[500, 140, 90, 15, 18, 13, 40, 2, 1, 7, 11, 12, 10, 33] - MaxHeap

[140, 33, 90, 15, 18, 13, 40, 2, 1, 7, 11, 12, 10 / 500]

[90, 33, 40, 18, 13, 15, 2, 1, 7, 11, 12, 10 / 500, 140]

[40, 33, 18, 13, 15, 2, 1, 7, 11, 12, 10 / 500, 140, 90]

[33, 18, 13, 15, 10, 1, 7, 11, 12, 2 / 500, 140, 90, 40]

[18, 13, 15, 12, 1, 7, 11, 10, 2 / 500, 140, 90, 40. 33]

[15, 13, 12, 2, 7, 11, 10, 1 / 500, 140, 90, 40. 33, 18]

[13, 12, 10, 7, 11, 2, 1 / 500, 140, 90, 40. 33, 18, 15]

[12, 11, 7, 10, 2, 1 / 500, 140, 90, 40. 33, 18, 15, 13]

[11, 7, 10, 2, 1 / 500, 140, 90, 40. 33, 18, 15, 13, 12]

[10, 7, 2, 1 / 500, 140, 90, 40. 33, 18, 15, 13, 12, 11]

[7, 2, 1 / 500, 140, 90, 40. 33, 18, 15, 13, 12, 11, 10]

[2, 1 / 500, 140, 90, 40. 33, 18, 15, 13, 12, 11, 10, 7]

[1 / 500, 140, 90, 40. 33, 18, 15, 13, 12, 11, 10, 7, 2]

[500, 140, 90, 40. 33, 18, 15, 13, 12, 11, 10, 7, 2, 1]

Explicación:

Se construyo una maxHeap usando el algoritmo BuildHeap. Luego se intercambia el primero con el ultimo y se reduce en uno el indice del arreglo, se verifica si se cumple la propiedad de maxHeap, caso contrario se filtranlos elementos. Se repite hasta que ya no quedan elementos en la maxHeap.

Ejercicio 8

Construir una max-heap binaria con los siguientes datos:

{5, 8, 12, 9, 7, 10, 21, 6, 14, 4}

a) Insertándolos de a uno

b) Usando el algoritmo BuildHeap

Respuesta:

a) [5]

[8, 5]

[12, 5, 8]

[12, 9, 8, 5]

[12, 9, 8, 5, 7]

[12, 9, 10, 5, 7, 8]

[21, 9, 12, 5, 7, 8, 10]

[21, 9, 12, 6, 7, 8, 10, 5]

[21, 14, 12, 9, 7, 8 ,10, 5, 6]

[21, 14, 12, 9, 7, 8, 10, 5, 6, 4]

b) [5, 8, 12, 9, 7, 10, 21, 6, 14, 4]

[5, 8, 12, 14, 7, 10, 21, 6, 9, 4]

[5, 8, 21, 14, 7, 10, 12, 6, 9, 4]

[5, 14, 21, 9, 7, 10, 12, 6, 8, 4]

[21, 14, 12, 9, 7, 10, 5, 6, 8, 4]

Explicación:

Mientras se cumpla la propiedad de maxHeap el arreglo es valido aunque ambas formas de construirlo generen ordenes distintos de los elementos.

Ejercicio 9

Suponga que una heap que representa una cola de prioridades está almacenada en el arreglo A (se comienza de la posición A[1]). Si insertamos la clave 16, ¿en qué posición quedará?

i:	1	2	3	4	5	6	7	8	9	10	11	12
A[i]:	11	21	27	37	36	34	32	43	44	42	51	62

(a) A[2] (b) A[3] (c) A[6] (d) A[7] (e) A[12]

Respuesta:

b

Explicación:

Al hacer el filtrado para restaurar la propiedad minHeap se intercambia el 16 con el 34 y luego con el 27, la heap final es [11, 21, 16, 37, 36, 27, 32, 43, 44, 42, 51, 62, 34]

Ejercicio 10

Suponga que una heap que representa una cola de prioridades está almacenada en el arreglo A (se comienza de la posición A[1]). Si aplica un DeleteMin, ¿en qué posición quedará la clave 62?

i:	1	2	3	4	5	6	7	8	9	10	11	12
A[i]:	11	21	27	37	36	34	32	43	44	42	51	62

(a) A[1] (b) A[2] (c) A[10] (d) A[11] (e) A[12]

Respuesta:

c

Explicación:

Luego de eliminar la raiz, el 62 queda al inicio del arreglo ya que era el ultimo elemento. Luego de filtrar hacia abajo para restaurar la propiedad, quedaria en la posicion 10. El arreglo final seria [21, 36, 27, 37, 42, 34, 32, 43, 44, 62, 51]

Ejercicio 11

a) Ordenar de forma creciente los datos del ejercicio anterior, usando el algoritmo HeapSort.
b) Cuales serian los pasos a seguir si se quiere ordenar de forma decreciente?

Respuesta:

a) [21, 27, 37, 36, 34, 32, 43, 44, 42, 51, 62] - [11]

[27, 36, 32, 37, 42, 34, 51, 43, 44, 62] - [11, 21]

[32, 36, 34, 37, 42, 62, 51, 43, 44] - [11, 21, 27]

[34, 36, 44, 37, 42, 62, 51, 43] - [11, 21, 27, 32]

[36, 37, 44, 43, 42, 62, 51] - [11, 21, 27, 32, 34]

[37, 42, 44, 43, 51, 62] - [11, 21, 27, 32, 34, 36]

[42, 43, 44, 62, 51] - [11, 21, 27, 32, 34, 36, 37]

[43, 51, 44, 62] - [11, 21, 27, 32, 34, 36, 37, 42]

[44, 51, 62] - [11, 21, 27, 32, 34, 36, 37, 42, 43]

[51, 62] - [11, 21, 27, 32, 34, 36, 37, 42, 43, 44]

[62] - [11, 21, 27, 32, 34, 36, 37, 42, 43, 44, 51]

[ ] - [11, 21, 27, 32, 34, 36, 37, 42, 43, 44, 51, 62]

b) Se construye un maxHeap, se intercambia el primero con el ultimo dentro del mismo arreglo y se reduce el indice en uno, se filtran los elementos restantes si es necesario y se repite hasta haber intercambiado todos los elementos del arreglo.

Explicación:

a) Se construye un minHeap y se aplica DeleteMin repetidamente hasta vaciarlo, en cada iteracion se restaura si es necesario la propiedad minHeap. Se agrega el elemento a un nuevo arreglo.
b) NA

Ejercicio 12

¿Cuáles de los siguientes arreglos representan una max-heap, min-heap o ninguna de las dos?

arreglo 1: 0 1 2 0 4 5 6 7 8 9

arreglo 2: 9 8 7 6 5 4 3 2 1 0

arreglo 3: 5 5 5 6 6 6 6 7 7 1

arreglo 4: 9 3 9 2 1 6 7 1 2 1

arreglo 5: 8 7 6 1 2 3 4 2 1 2

Respuesta:

MaxHeap: arreglo 2 y arreglo 4

MinHeap: ninguno

Ninguno: arreglo 1, arreglo 3 y arreglo 5

Explicación:

Los arreglos 2 y 4 cumplen la propiedad maxHeap. Ninguno cumple la propiedad minHeap y los arreglos 1, 3 y 5 no cumplen ninguna de las dos.

Ejercicio 13

Un arreglo de 7 enteros se ordena ascendentemente usando el algoritmo HeapSort. Luego de la fase inicial del algoritmo (la construcción de la heap), ¿cuál de los siguientes es un posible orden del arreglo?

(a) 85 78 45 51 53 47 49

(b) 85 49 78 45 47 51 53

(c) 85 78 49 45 47 51 53

(d) 45 85 78 53 51 49 47

(e) 85 51 78 53 49 47 45

Respuesta:

b

Explicación:

Es el unico maxHeap.

Ejercicio 14

En una Heap, ¿para un elemento que está en la posición i su hijo derecho está en la posición.....?

(a) | i/2 |

(b) 2*i

(c) 2*i + 1

(d) Ninguna de las anteriores

Respuesta:

c

Explicación:

NA

Ejercicio 15

¿Siempre se puede decir que un árbol binario lleno es una Heap?

(a) Sí (b) No

Respuesta:

b

Explicación:

Un heap es un arbol binario completo, no lleno.

Ejercicio 16

La operación que agrega un elemento a la heap que tiene n elementos, en el peor caso es de .....

(a) O(n)

(b) O(n log n)

(c) O(log n)

(d) Ninguna de las otras opciones

Respuesta:

b

Explicación:

Agregar un elemento tiene un tiempo de O(log n) por lo tanto agregar n elementos tendra uno de O(n log n)

Ejercicio 17

Se construyó una Máx-Heap con las siguientes claves: 13, 21, 87, 30, 25, 22, 18. ¿Cuál de las siguientes opciones corresponde al resultado de realizar la construcción insertando las claves una a una?

a) 87, 30, 25, 22, 21, 18, 13
b) 87, 30, 22, 21, 25, 13, 18
c) 87, 30, 25, 13, 22, 18, 21
d) 87, 30, 22, 13, 25, 21, 18

Respuesta:

d

Explicación:

NA

Ejercicio 18

Se construyó una Máx-Heap con las siguientes claves: 13, 21, 87, 30, 25, 22, 18. ¿Cuál de las siguientes opciones corresponde al resultado de realizar la construcción aplicando el algoritmo Build-Heap?

a) 87, 30, 25, 22, 21, 18, 13
b) 87, 30, 22, 21, 25, 13, 18
c) 87, 30, 25, 13, 22, 18, 21
d) 87, 30, 22, 13, 25, 21, 18

Respuesta:

b

Explicación:

NA

Ejercicio 19

El algoritmo HeapSort consta de dos etapas:

1) se construye una heap

2) se realizan los intercambios necesarios para dejar ordenados los datos.

Asuma que la heap ya está construida y es la siguiente: 58 38 53 23 28 40 35 18

¿Cómo quedan los datos en el arreglo después de ejecutar sólo 2 pasos de la segunda etapa del

Heapsort?

(a) 40 38 23 28 35 18 53 58

(b) 53 38 40 23 28 18 35 58

(c) 40 38 23 35 28 18 53 58

(d) 40 38 35 23 28 18 53 58

Respuesta:

d

Explicación:

NA

Ejercicio 20

Dada la Min-Heap 3, 8, 5, 15, 10, 7, 19, 28, 16, 25, 12. ¿En qué posición está ubicado el hijo derecho de la clave 15?

(a) 7

(b) 8

(c) 9

(d) 10

Respuesta:

c

Explicación:

2*4(indice de 15) + 1 = 8 + 1 = 9

Ejercicio 21

Construya una min-heap con las siguientes claves: 15, 25, 23, 13, 18, 2, 19, 20, 17 insertándose una a una. Indique en qué posiciones quedaron ubicadas las claves: 2, 18 y 25.

Respuesta:

[15]

[15, 25]

[15, 25, 23]

[13, 15, 23, 25]

[13, 15, 23, 25, 18]

[2, 15, 13, 25, 18, 23]

[2 ,15, 13, 25, 18, 23, 19]

[2, 15, 13, 20, 18, 23, 19, 25]

[2, 15, 13, 17, 18, 23, 19, 25, 20]

Clave 2 en posición 1.

Clave 18 en posicion 5.

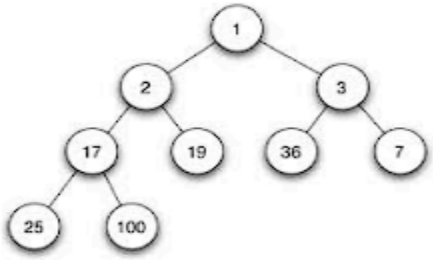
Clave 25 en posicion 8.

Explicación:

NA

Ejercicio 22

Luego de insertar la clave 15 en la siguiente min-heap, ¿cuántas de las claves que ya estaban en la heap han mantenido su lugar (es decir, ocupan en la min-heap resultante la misma posición que ocupaban antes de la inserción)?



a) Ninguna
b) Seis
c) Ocho
d) Nueve

Respuesta:

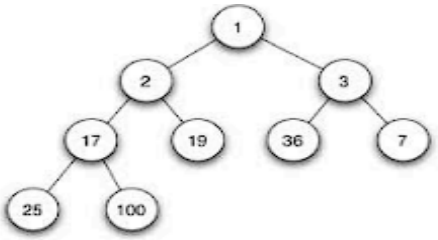
c

Explicación:

Al insertar la clave 15 su padre seria la clave nueve la cual es mayor, ocurre una filtracion para ajustar la propiedad y unicamente el elemento 19 es el que cambia de lugar dejando los otros ocho valores sin cambios.

Ejercicio 23

Luego de una operación de borrado del mínimo en la siguiente min-heap, ¿cuántas claves han cambiado de lugar (es decir, ocupan en la min-heap resultante un lugar diferente al que ocupaban en la min-heap antes del borrado) ? (No contar la clave borrada, ya que no pertenece más a la heap)



a) Ninguno
b) Dos
c) Tres
d) Cuatro

Respuesta:

b

Explicación:

Al borrar el minimo el valor 2 se convierte en la raiz, esto rompe la propiedad ya que ahora el 17 es padre del 7, hay que filtrar para ajustar asi que intercambian posiciones, el resto de valores permanece igual.