Practica 5

Nombre: Jorge Ignacio Heredia Bazoalto

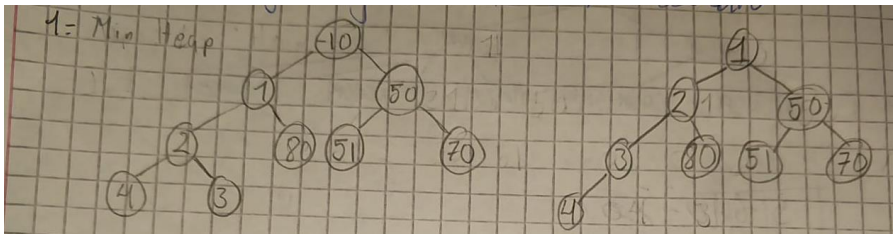
Crear los siguientes arboles Heaps representados gráficamente uno después del insert de todos los valores y otro después de la eliminación

1)

Tipo: min heap

Insertar: 1,2,50,3,80,51,70,4, -10

Eliminar: -10

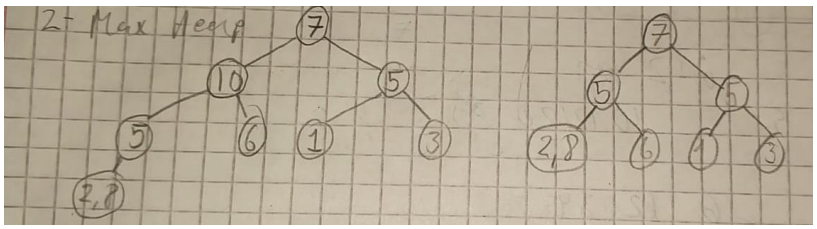


2)

Tipo: max heap

Insertar: 7,6,1,5,10,5,3,2.8

Eliminar: 10

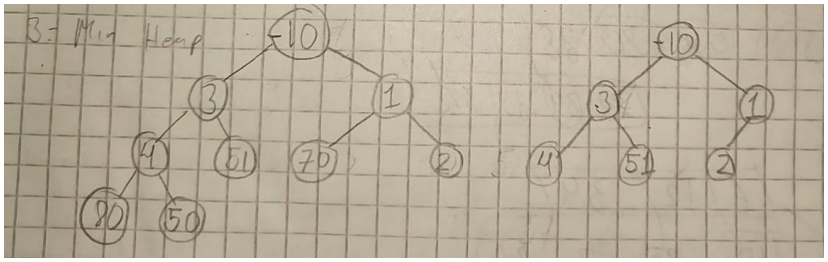


3)

Tipo: min heap

Insertar: 3, 1,2,80,51,70, -10,50,4

Eliminar: 80, 70, 50

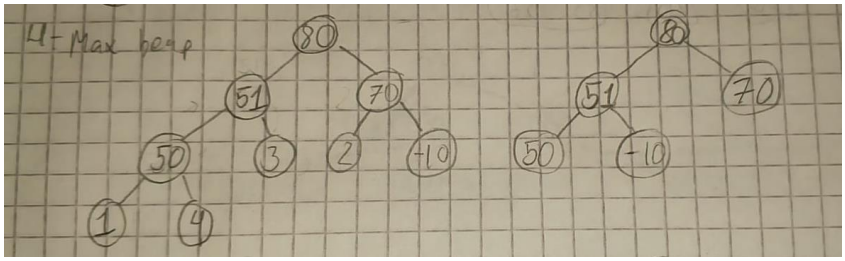


4)

Tipo: max heap

Insertar: 3, 1,2,80,51,70, -10,50,4

Eliminar: 1, 2, 4, 3

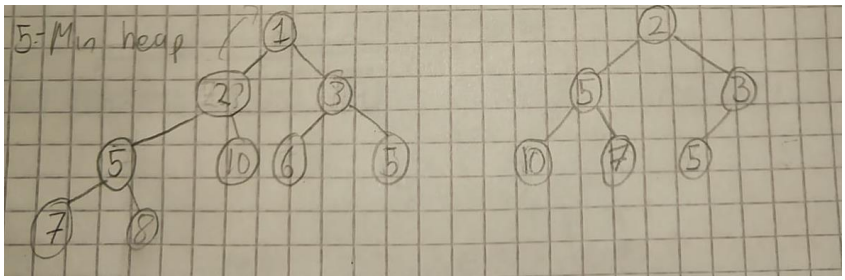


5)

Tipo: min heap

Insertar: 7,6,1,5,10,5,3,2.8

Eliminar: 6,1,8

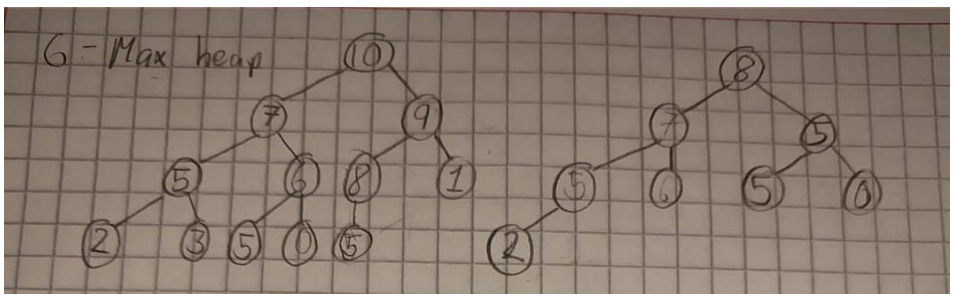


6)

Tipo: max heap

Insertar: 2,8,5, 7,6, 10,1,5,3,5, 0, 9

Eliminar: 1,10, 3, 9

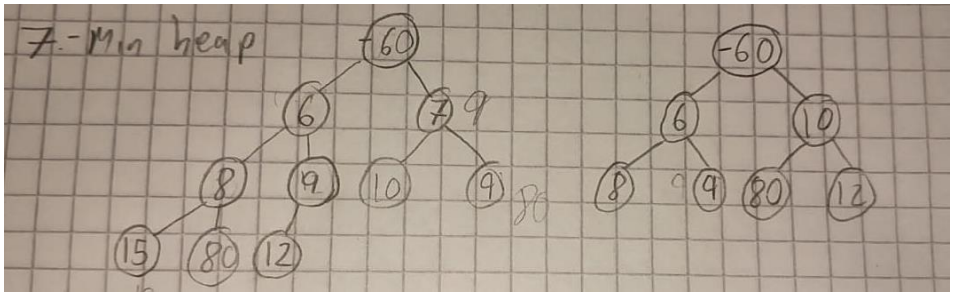


7)

Tipo: min heap

Insertar: 15, 10, 8, 12, 7, 6, 9,-60,80,9

Eliminar: 15, 7, 9

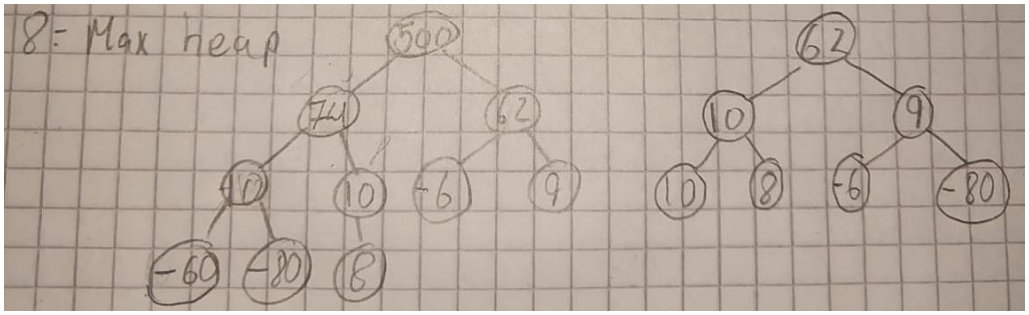


8)

Tipo: max heap

Insertar: 500, 10, 62, 74, 10,-6, 9,-60,-80, 8

Eliminar: 74,-60, 500



Teoría:

¿Que son las Colas de Prioridad y como se puede aplicar los árboles heap en estos?

Una cola de prioridad es un tipo de dato o estructura de datos, similar a una cola que además de almacenar el valor del nodo, cada nodo tiene una prioridad que lo hace mas prioritario a otros nodos. Entonces para aplicar ese concepto en los árboles heap, podemos ordenar cada elemento en base a su prioridad y por tanto tener arriba y más fácil de acceder los nodos con mayor prioridad.

¿Por qué se dice que heapsort es más rápido en algunos casos que Quicksort?

HeapSort tiene la ventaja de tener un tiempo de ejecución logarítmico en el peor caso O(n log n), en cambio QuickSort tiene un tiempo en el peor caso cuadrático $O(n^2)$.

¿Qué es el método buildheap y cómo funciona? de 2 ejemplos.

Es un algoritmo o método que nos permite crear un tree heap a partir de un array.

Por ejemplo:

[22, 4, 52, 75, 1, 45] Max Heap.

```
          [52]
   [75]          [45]
[4]    [1]   [22]   [ ]
```

[75, 3, 2, 7, 45, 9, 0] Min Heap.

```
          [0]
   [7]          [2]
[75]  [45]  [9]    [3]
```

¿Por qué se dice que la búsqueda en los árboles heap no es óptima y como se podría mejorar?

Los árboles heap están ordenados verticalmente, pero no horizontalmente, por lo que para buscar un elemento que esta abajo, hay que visitar varios antes de encontrarlo.
Para mejorarlo podríamos poner el hermano menor a la izquierda, y no admitir duplicados.

De 5 ejemplos de situaciones reales en donde se puede aplicar los árboles heap.

1. Planificación de eventos de un servidor.
2. Gestión de memoria en sistemas operativos
3. Gestión de tareas en un sistema operativo.
4. Colas de prioridad en un hospital.
5. Logistica de entrega para repartir paquetes.