Montículos

Máximos y mínimos

Ing. Miguel Matul Calderón

Introducción

Los montículos son estructuras de datos que nos ayudan a realizar eficientemente las operaciones de: encontrar el elemento **máximo/mínimo** dentro de una colección de elementos en un tiempo constante.

Esta estructura de datos fue propuesta por Robert W. Floyd (premio Turing en 1978) para resolver el problema de la ordenación de elementos dentro de un vector, el famoso heapsort (u ordenación por montículo).

Montículo

Un montículo es en esencia un **árbol binario balanceado** que cumple con la premisa de que: ningún padre tiene un hijo mayor (montículo de máximos) o menor (montículo de mínimos) a él.

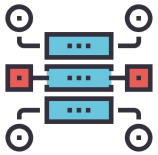
Hay dos tipos principales:

- Montículo mínimo (Min Heap)
- Montículo máximo (Max Heap)

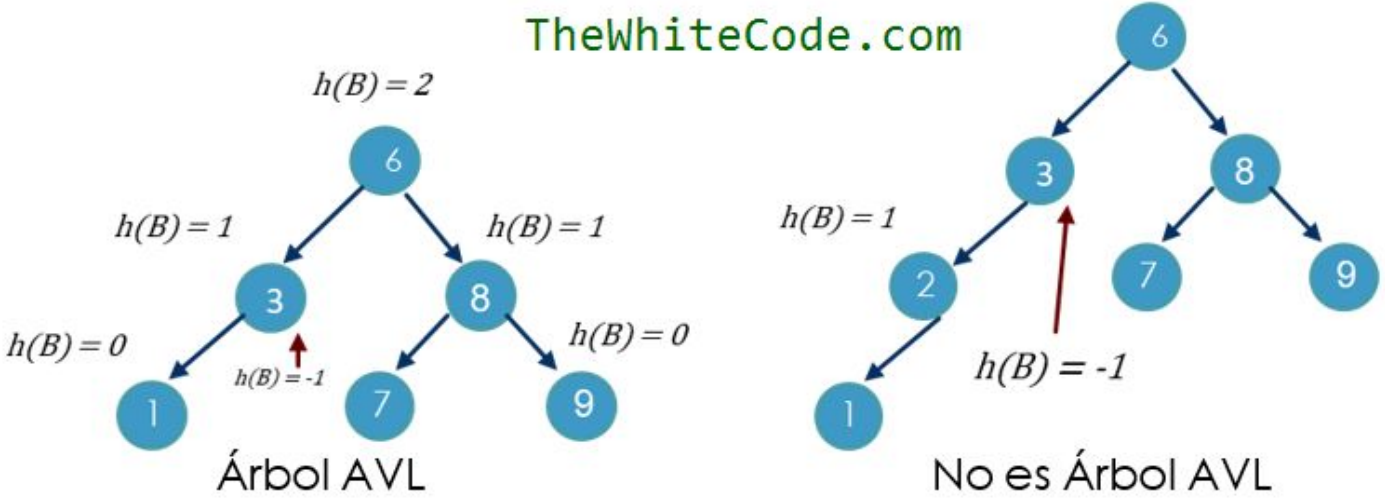
Árbol binario balanceado

Un árbol binario balanceado **(AVL)** es un árbol binario donde todos los niveles están completamente llenos, con excepción del último nivel.

En un árbol binario balanceado el último nivel tiene nodos de tal manera que el lado izquierdo nunca está vacío.

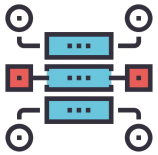


Árbol binario balanceado

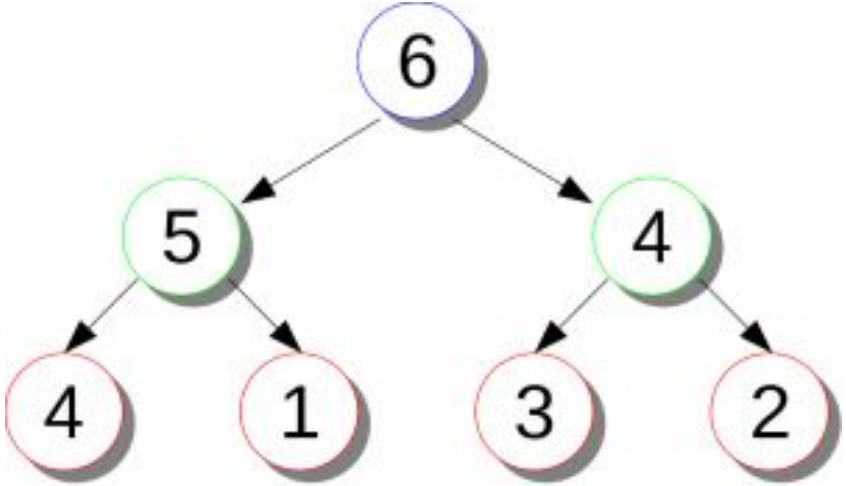


Representación de un montículo

- Usualmente se implementan usando arreglos
- La primera entrada de un arreglo se toma como vacía
- Como los montículos son árboles binarios balanceados, los valores son almacenados en un arreglo modelando nivel por nivel de izquierda a derecha.



Representación de un montículo



Aplicaciones

El montículo permite:

- Ordenar elementos de un vector de forma fácil y óptima.
- Búsquedas de máximos, mínimos, o cualquier otro factor a tener en cuenta en la selección de búsqueda de elementos.
- En problemas específicos de grafos como el camino más corto de Dijkstra.

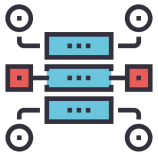
Fórmulas

La fórmula para acceder a cada hijo, dado el índice del padre (i), sería:

- hijo_izquierdo = 2i
- hijo_derecho = 2i + 1

El acceso al padre se realizaría con una división entera:

- padre = i / 2.



Pseudocódigo Max Heap

```
// atributos
entero[] monticulo
entero n

// métodos
constructor(entero: capacidad) {
        monticulo = nuevo Arreglo(capacidad + 1)
        n = 0
}
función booleana estaVacia() {
        regresar n == 0
}
función entera obtenerTamaño() {
        regresar n
}
```

Pseudocódigo Max Heap

```
procedimiento insertar(entero: valor) {
        si (n == monticulo.longitud() - 1)
                redimensionar(2 * monticulo.longitud()
        n++
        monticulo[n] = valor
        intercambiar(n)
}
procedimiento intercambiar(entero: i) {
        entero padre = i / 2
        mientras (i > 1 && monticulo[padre] < monituclo[i]) {
                entero temp = monticulo[i]
                monticulo[i] = monticulo[padre]
                monticulo[padre] = temp
                i = i / 2
                padre = i / 2
        }
}
```

Ejercicio

Construya el montículo de acuerdo al siguiente orden de inserción de elementos:

- 4, 5, 2, 6, 1, 3, 9
- ¿Cómo se dibuja este montículo?
- ¿Cómo se compone el arreglo que representa al montículo?

Montículos

Máximos y mínimos

Ing. Miguel Matul Calderón