

## MONTÍCULOS SESGADOS

### INTRODUCCIÓN

Los montículos sesgados son árboles binarios que satisfacen la propiedad de un montículo: bien es vacío, o bien en la raíz se encuentra el elemento mínimo, y adicionalmente sus hijos son montículos sesgados.

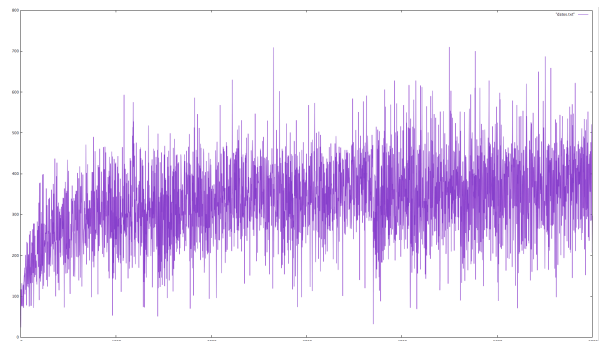
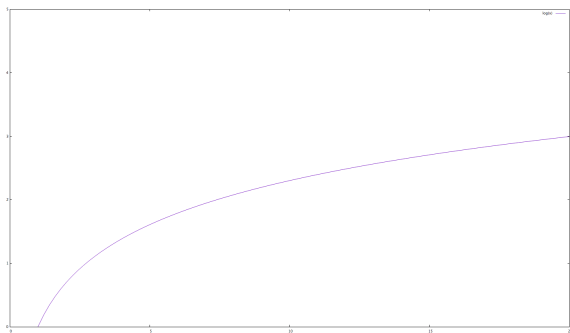
Las operaciones realizadas para este montículo han sido: unión de dos montículos, inserción de un elemento en un montículo, obtención del elemento mínimo del montículo, borrado del mínimo elemento del montículo y modificación de un elemento (mayor o menor que el actual) del montículo, por otro.

En todas las operaciones excepto en la obtención del mínimo, se hace uso de la unión de montículos sesgados, por lo que los costes son similares.

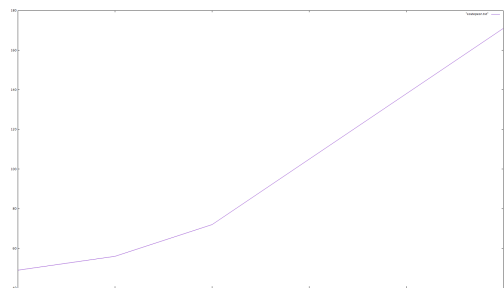
### OPERACIONES Y GRÁFICAS CON TIEMPOS

#### Unión:

La unión de dos montículos tiene un coste  $O(\log n)$ , siendo  $n$  el número de nodos. En la gráfica podemos apreciar como los datos recogidos fluctúan con valores mayores y menores y a medida que aumenta el número de nodos se va formando una función que crece de manera similar a la del  $\log n$ .

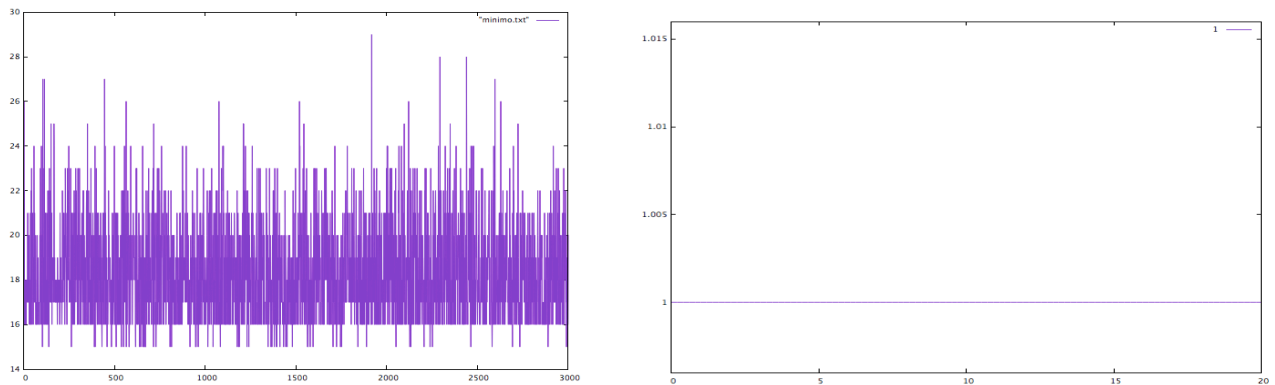


El caso peor de la unión de dos montículos sesgados es aquel en el que ambos montículos únicamente poseen hijos derechos, formando una especie de “espina”. En este caso, la complejidad está en  $O(n_1+n_2)$ , siendo  $n_1$  el número de nodos de un montículo y  $n_2$  el del otro. Puede observarse en la siguiente gráfica como el tiempo aumenta a medida que aumenta el número de nodos.



### Mínimo:

El cálculo del mínimo tiene un coste  $O(1)$ , ya que el número de nodos u otros factores no afectan al coste. Dadas las propiedades de un montículo sesgado, únicamente es necesario acceder al nodo raíz y tomar su elemento, que será el mínimo. Puede observarse en la siguiente gráfica como es prácticamente constante.



### Inserción:

La inserción consiste en la unión entre el montículo donde se quiere insertar y el nuevo nodo (convertido en un montículo de un solo nodo).

La gráfica sería muy parecida a la de la unión, ya que es una unión normal lo que realmente se realiza. Esto es lo mismo que sucede con el resto de operaciones del montículo sesgado, todas utilizan una o más veces la unión, con lo que no es necesario realizar gráficas para ellas.

### Borrado del mínimo:

El borrado del mínimo se realiza eliminando la raíz del montículo y uniendo sus hijos derecho e izquierdo de manera que formen un nuevo montículo.

### Modificar una clave:

La modificación de la clave en el código está implementada de tal manera que siempre el nodo modificado es el hijo izquierdo del hijo izquierdo de la raíz, pero puede ser cambiado desde el código por cualquier otro nodo existente. El algoritmo para la modificación de la clave se realiza de la siguiente manera: se elimina del montículo el nodo a cambiar, no sin antes guardar sus montículos hijos derecho e izquierdo y se realiza la unión de éstos; por último, se une la unión anterior con el árbol original y se inserta el nuevo valor.

## CASOS DE PRUEBA SENCILLOS DE LAS OPERACIONES

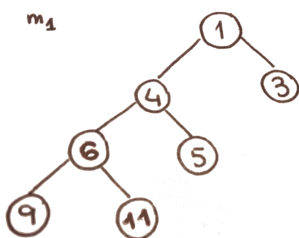
A continuación, se muestran las entradas y salidas de algunos ejemplos para las distintas operaciones.

### Unión:

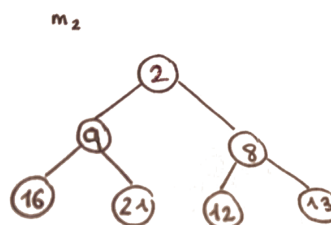
Sea  $m_1$  el primer montículo a unir y  $m_2$  el segundo montículo.

#### Entrada

$$m_1 = ((([9]6[11])4[5])1[3])$$

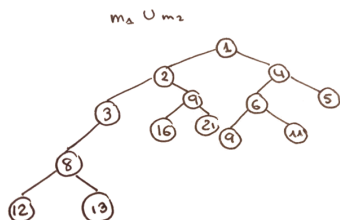


$$m_2 = ((([16]9[21])2([12]8[13]))$$



#### Salida

$$m_1 \cup m_2 = (((([12]8[13])3\#)2([16]9[21]))1([9]6[11])4[5]))$$



### Mínimo:

Sea  $m$  el montículo del cual se quiere sacar el mínimo.

#### Entrada

$$m = ([7]2[9])$$



#### Salida

El mínimo del montículo es: 2

### **Inserción:**

Sea  $m$  el montículo donde se quiere insertar,  $e$  el elemento a insertar y  $m'$  el montículo final.

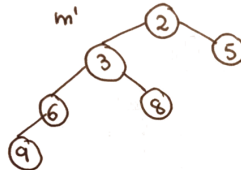
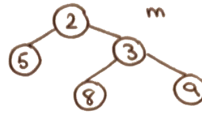
#### Entrada

$m = ([5]2([8]3[9]))$

$e = 6$

#### Salida

$m' = ((([9]6\#)3[8])2[5])$



### **Borrado del mínimo:**

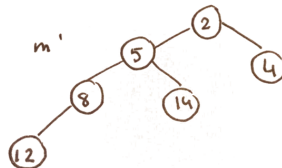
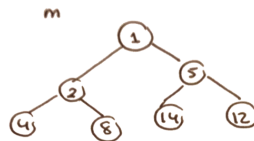
Sea  $m$  el montículo del cual se va a borrar el mínimo (la raíz) y  $m'$  el montículo ya sin el mínimo.

#### Entrada

$m = ((([4]2[8])1([14]5[12])))$

#### Salida

$m' = ((([12]8\#)5[14])2[4])$



### **Modificar una clave:**

Sea  $m$  el montículo en el que cambiar un nodo (en este caso el hijo izquierdo del hijo izquierdo de la raíz),  $v$  el valor por el que se va a cambiar y  $m'$  el montículo resultante.

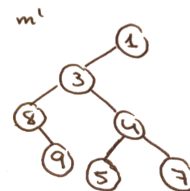
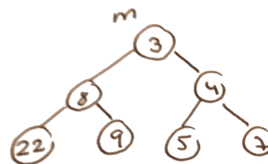
#### Entrada

$m = ((([22]8[9])3([5]4[7])))$

$v = 1$

#### Salida

$m' = (((\#8[9])3([5]4[7]))1\#)$



## INFORMACIÓN SOBRE LA ENTRADA

Los montículos se codifican en la entrada de acuerdo con el siguiente criterio:

- El montículo vacío se representa como #
- Un montículo simple con raíz A se representa como [A]
- Un montículo compuesto con raíz A se representa como  $(\tau_i A \tau_d)$ , donde  $\tau_i$  es la representación del hijo izquierdo, y  $\tau_d$  la del derecho.

## INFORMACIÓN SOBRE LA EJECUCIÓN

El código contiene 3 main distintos:

1. MainManual: Contiene un menú e indicaciones de lo que se tiene que ir introduciendo (con el formato de entrada anterior descrito), para realizar las operaciones.
2. MainTiempos: Implementado para la realización de gráficas.