Laboratorio 2

Juan Diego Sique Martínez

Agosto 2018

1. Primer ejercicio

El arreglo original es éste.

[3, 39, 61, 91, 57, 22, 75, 89, 9, 90, 63, 78, 28, 73, 20]

El arreglo ordenado queda así:

[91, 90, 89, 78, 75, 73, 63, 61, 57, 39, 28, 22, 20, 9, 3]

Entra al método MergeSort **29** veces. Por lo tanto la relación entre el número de ítems es **1.9333333333333333**

2. Segundo ejercicio

Para demostrar el tiempo de ejecución de mi algoritmo se puede demostrar mediante las propiedades de un «heap». La primera propiedad es la que establece que un heap debe de tener dos hijos menores que éste (asumiendo que es un «max-heap», por lo contrario sería un «min-heap») y cada elemento de la lista debe de poseer ésta propiedad de manera recursiva. En el caso de las hojas, se consideran un «heap». Por lo tanto debe de recorrer todos los elementos para verificar que cumplan con la propiedad. Por lo tanto se ejecuta $O(n)$.

3. Tercer ejercicio

Data: Un arreglo **A**, un entero **i** que representa el índice del arreglo
Result: Una corrección de las ramas del arreglo donde se corrige para que cumpla con la propiedad de un «heap»

```
initialization;
while i > -1 do
    l ← LEFT(i);
    r ← RIGHT(i);
    if l ≤ heap.size(A) and A[l] > A[i] then
        largest ← l;
    else
        largest ← i;
    end
    if r ≤ heap.size(A) and A[r] > A[largest] then
        largest ← r;
    end
    if largest ≠ i then
        exchange A[largest] ↔ A[i];
        i ← largest;
    else
        i ← -1
    end
end
```

Algorithm 1: Max-Heapify

La única variación que el código sufrió es el intercambio de una instrucción «while» por la recursividad donde el valor de **i** es el que se utiliza como comodín entre ambas técnicas. Para ello, bastaba entender que si se caía en la función nuevamente se debe de cumplir la condición valor del elemento en el arreglo con índice **i** era diferente al que tiene el índice **largest**. Si no es así romper la instrucción «while» haciendo que no se cumpla la condición es decir, asignando el valor -1(o cualquier otro negativo) a **i**.