

Lab1

Fernando González

9 de agosto de 2018

Problema 1

Yo utilice el algoritmo de ordenamiento Merge Sort, el cual tiene un *running time* de $O(n\log(n))$. Mi código ordena el arreglo dado gracias a la combinación de 3 funciones las cuales se llaman remisivamente, permitiendo partir el arreglo en sub-arreglos y así poder arreglarlo de forma ordenada de mayor a menor. En total, dado mis 3 funciones, mi programa se itero 44 veces según mi contador de funciones.

Problema 3

Algorithm 1 Max Heapify Iterativo

```
1: procedure MHEAPIFY(A,i)
2:   Initialization:
3:   A[x..]
4:   while  $i \leq \text{lean}(A)$  do
5:      $l = \text{Left}[i]$ 
6:      $r = \text{Right}[i]$ 
7:     if  $A[l] \geq A[i]$  then
8:       Return  $\text{largest} = l$ 
9:     else Return  $\text{largest} = i$ 
10:    if  $A[r] \geq A[\text{largest}]$  then
11:      Return  $\text{largest} = r$ 
12:    if  $\text{largest} \neq i$  then
13:      Return Swift  $A[i] < - > A[\text{largest}]$ 
14:       $i = \text{largest}$ 
15:    else Return 0
    Left(i)  $\rightarrow (2i)$ 
    Right(i)  $\rightarrow 2i + 1$ 
```
