



Algoritmos y Programación

Práctica 5.2: Merge Sort

Merge Sort

Programa en Python el algoritmo de ordenación por fusión (*merge sort*)

```
MergeSort (data, left, right) {  
    if (left < right) {  
        mid = divide (left, right)  
        MergeSort (data, left, mid)  
        MergeSort (data, mid+1, right)  
        Merge (data, left, mid+1, right)  
    }  
}
```

Pseudocode for Merge:

C = output [length = n]	for k = 1 to n
A = 1 st sorted array [n/2]	if A(i) < B(j)
B = 2 nd sorted array [n/2]	C(k) = A(i)
i = 1	i++
j = 1	else [B(j) < A(i)]
	C(k) = B(j)
	j++
	end
	(ignores end cases)

Formato del fichero de entrada

- La primera línea es un descriptor con el número de valores a ordenar
- El resto de las líneas son los valores a ordenar

Ejemplo:

Input = 10

1

9

4

9

9

4

0

5

9

0

Output = [0, 0, 1, 4, 4, 5, 9, 9, 9, 9]

VPL

main.py

```
1 from solve import *
2
3 first_line = input().split()
4 numValues = int(first_line[0])
5
6 items = []
7 for j in range(1, numValues+1):
8     parts = input().split()
9     items.append(int(parts[0]))
10
11 solve(items)
12 print(items)
```

solve.py

```
1 def solve(items):
2     """
3     Sort the given list of items in ascending order
4     """
5
6     def merge(left, mid, right):
7         # solve it here!
8         ...
9
10
11     def merge_sort(left, right):
12         # solve it here!
13         ...
14
15
16     merge_sort(0, len(items)-1)
17     return
18
```

Este ejercicio no puntúa