uc3m | Universidad Carlos III de Madrid

OpenCourseWare

Tema 7 - Divide y Vencerás

Hoja de Problemas

1. Implementa una función, find_max, que reciba una lista de números y devuelva el mayor. La solución debe estar basada en divide y vencerás.
    a. Implementa una primera versión en la que se utilice slicing para la lista.
    b. Implementa una segunda versión que mejore la complejidad espacial (es decir, que no esté basada en slicing).

   Solución: función find_max y find_max2 en tema7_problemas.py
2. Implementa una función, find_min_max, que reciba una lista de números y devuelva una tupla formada por el elemento más pequeño y el mayor de la lista. La solución debe estar basada en divide y vencerás.
   Solución: función find_min_max en tema7_problemas.py
3. Implementa una función, find_lowest_even_odd, que reciba una lista de números y devuelva una tupla formada por el par y el impar más pequeños de la lista. La solución debe estar basada en divide y vencerás.
   Solución: función find_lowest_even_odd en tema7_problemas.py
4. Implementa una función, sum_list, que reciba una lista de números y devuelva la suma de todos sus elementos.
   Solución: función sum_list en tema7_problemas.py
5. Implementa una función, sum_multiple_5, que reciba una lista de números y devuelva la suma de todos los múltiplos de 5 que ocurran en la lista.
   Solución: función sum_multiple_5 en tema7_problemas.py
6. Implementa una función, get_words_len_lower_2, que reciba una lista de palabras y devuelva una nueva lista con las palabras con longitud igual o menor a 2.
   Solución: función get_words_len_lower_2 en tema7_problemas.py
7. Implementa el algoritmo mergesort. La función recibe una lista y devuelve una nueva lista ordenada.
    a. Recuerda que tienes que implementar una función auxiliar que tome dos listas ordenadas y devuelva una lista que contenga a los elementos de ambas listas, ordenados de menor a mayor.

8. Implementa una versión del algoritmo quicksort donde el pivote sea el elemento central. En esta implementación, la función quicksort toma una

lista de entrada y la ordena. Es decir, la función no devuelva una nueva lista, sino que modifica la lista de entrada.

9. Implementa una versión del algoritmo quicksort, quicksort_random, donde el pivote sea un elemento elegido al azar entre los elementos de la lista. En esta implementación, la función quicksort toma una lista de entrada y la ordena. Es decir, la función no devuelva una nueva lista, sino que modifica la lista de entrada.
10. Implementa una función quicksort_first que reciba una lista y devuelva una nueva lista con los elementos de la lista de entrada ordenados de forma ascendente. La función debe estar basada en quicksort, tomando como pivote el primer elemento de la lista.
11. Implementa una función quicksort_last que reciba una lista y devuelva una nueva lista con los elementos de la lista de entrada ordenados de forma ascendente. La función debe estar basada en quicksort, tomando como pivote el último elemento de la lista.