Домашняя работа от 07.03.2024.  
Выполнил Суганов Борис

Группа МИСТ 23-3-3

**Задание 2\*\***. Для обоих видов сортировки рассмотрите случаи, когда в массиве есть отрицательные элементы

**Код(LSD):**

def task\_1\_1(arr):  
 print(f'Исходный список: {" ".join(str(el) for el in arr)}')  
 result = lsd\_sort(arr)  
  
 print(f'Результат: {" ".join(str(el) for el in [int(x) for x in result])}', '\n')  
 return result  
  
def lsd\_sort(arr):  
 if not arr:  
 return []  
 d = []  
 m = []  
  
 max\_length = len(str(abs(max(arr, key=abs))))  
 arr\_s = fill\_array\_with\_zeros(arr, max\_length)  
  
 for r in range(1, max\_length + 1):  
 for i in range(-9, 10):  
 d = spread\_numbers\_into\_bins(arr\_s, i, r)  
 m += d  
 d.clear()  
  
 arr\_s = m.copy()  
 m.clear()  
  
 return [int(x) for x in arr\_s]  
  
  
def fill\_array\_with\_zeros(arr, max\_length):  
 result\_arr = []  
 for num in arr:  
 if num >= 0:  
 processed\_num = str(num).zfill(max\_length)  
 else:  
 processed\_num = '-' + str(abs(num)).zfill(max\_length)  
 result\_arr.append(processed\_num)  
 return result\_arr  
  
  
def spread\_numbers\_into\_bins(arr, bin, rank):  
 result\_arr = []  
 for x in arr:  
 coef = '-'  
 if int(x) >= 0 or bin == 0:  
 coef = ''  
 if (coef + x[-rank]) == str(bin):  
 result\_arr.append(x)  
 return result\_arr

Результаты:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Код(MSD)** (на след странице)

def task\_1\_2(arr):  
 print(f'Исходный список: {" ".join(str(el) for el in arr)}')  
 result = radix(arr)  
 print(f'Результат: {" ".join(str(el) for el in result)}', '\n')  
 return [int(x) for x in result]  
  
def radix(array, size=None, index=None):  
 if not array:  
 return array  
 if size is None:  
 size = len(str(abs(max(array, key=abs))))  
 if index is None:  
 index = size  
 if index <= 0:  
 return array  
  
 bins = [[] for \_ in range(19)]  
  
 for e in array:  
 coef = 1  
 if e < 0:  
 coef = -1  
  
 num\_with\_zeros = str(abs(e)).zfill(size)  
 dest\_c = num\_with\_zeros[size - index]  
 dest\_i = int(dest\_c)  
 bins[9 + coef \* dest\_i ] += [e]  
  
 result = []  
 for b in bins:  
 if b:  
 result.append(radix(b, size, index - 1))  
  
 flattened\_result = flatten(result)  
 return flattened\_result  
  
  
def flatten(array):  
 new\_array = []  
 for sub\_array in array:  
 new\_array += sub\_array  
 return new\_array

Изображение выглядит как текст, снимок экрана, Шрифт, алгебра

Автоматически созданное описаниеРезультаты