

Тренировки по алгоритмам 4.0 от Яндекса — Занятие 1 (Сортировки: быстрая, слиянием и поразрядная)

🕒 2 апр 2024, 06:26:41
старт: 1 ноя 2023, 21:00:00
...

Объявления жюри

[Положение участников](#) [Задачи](#) [Посылки](#)

С. Слияние

Ограничение времени	5 секунд
Ограничение памяти	512Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

- ✓

A. Partition
- ✓

B. Быстрая сортировка
- ✓

C. Слияние
- ✓

D. Сортировка слиянием
- ✓

E. Поразрядная сортировка

Базовый алгоритм для сортировки слиянием — алгоритм слияния двух упорядоченных массивов в один упорядоченный массив. Эта операция выполняется за линейное время с линейным потреблением памяти. Реализуйте слияние двух массивов в качестве первого шага для написания сортировки слиянием.

Формат ввода

В первой строке входного файла содержится число N — количество элементов первого массива ($0 \leq N \leq 10^6$).
Во второй строке содержатся N целых чисел a_i , разделенных пробелами, отсортированные по неубыванию ($-10^9 \leq a_i \leq 10^9$).
В третьей строке входного файла содержится число M — количество элементов второго массива ($0 \leq M \leq 10^6$).
В третьей строке содежатся M целых чисел b_i , разделенных пробелами, отсортированные по неубыванию ($-10^9 \leq b_i \leq 10^9$).

Формат вывода

Выведите результат слияния этих двух массивов, то есть $M + N$ целых чисел, разделенных пробелами, в порядке неубывания.

Пример 1

Ввод 	Вывод 
5 1 3 5 5 9 3 2 5 6	1 2 3 5 5 5 6 9

Пример 2

Ввод 	Вывод 
1 0 0	0

Пример 3

Ввод 	Вывод 
0 1 0	0

Примечания

Для решения этой задачи советуем реализовать функцию, которая принимает на вход две пары итераторов, задающие два массива, и итератор на начало буфера, в который необходимо записывать результат. Итераторы можно заменить на передачу массивов и индексов в них. В таком виде вам будет удобно использовать эту функцию для реализации сортировки.

Язык Python 3.9 (PyPy 7.3.11) ▾

Набрать здесь Отправить файл

```
1 fin = open('input.txt')
2 N = int(fin.readline())
3 arr1 = [int(x) for x in fin.readline().split()]
4 M = int(fin.readline())
5 arr2 = [int(x) for x in fin.readline().split()]
6
7 def merge(array1, left1, right1, array2, left2, right2):
8     res = []
9
10    curr1, curr2 = left1, left2
11    while (curr1 <= right1) and (curr2 <= right2):
12        if array1[curr1] <= array2[curr2]:
13            res.append(array1[curr1])
14            curr1 += 1
15        else:
16            res.append(array2[curr2])
17            curr2 += 1
18
19    while curr1 <= right1:
20        res.append(array1[curr1])
21        curr1 += 1
22
23    while curr2 <= right2:
24        res.append(array2[curr2])
25        curr2 += 1
26
27    return res
28
29 print(*merge(arr1, 0, len(arr1)-1, arr2, 0, len(arr2)-1))
```

Отправить 📄 осталось 99 попыток

Предыдущая Следующая

Время посылки	ID	Задача	Компилятор	Вердикт	Тип посылки	Время	Память	Тест	Баллы
3 ноя 2023, 02:38:30	95729395	C	Python 3.9 (PyPy 7.3.11)	OK	-	1.342s	294.30Mb	-	- отчёт