

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

2 апр 2024, 06:27:00  
старт: 1 ноя 2023, 21:00:00  
...

Объявления жюри

Положение участников Задачи Посылки

## Е. Поразрядная сортировка

Ограничение времени	1 секунда
Ограничение памяти	64Mb
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

Поразрядная сортировка является одним из видов сортировки, которые работают практически за линейное от размера сортируемого массива время. Такая скорость достигается за счет того, что эта сортировка использует внутреннюю структуру сортируемых объектов. Изначально этот алгоритм использовался для сортировки перфокарт. Первая его компьютерная реализация была создана в университете MIT Гарольдом Сьюардом (Harold H. Seward). Опишем алгоритм подробнее. Пусть задан массив строк  $s_1, ..., s_i$  причём все строки имеют одинаковую длину  $m$ . Работа алгоритма состоит из  $m$  фаз. На  $i$ -ой фазе строки сортируются па  $i$ -ой с конца буквы. Происходит это следующим образом. Будем, для простоты, в этой задаче рассматривать строки из цифр от 0 до 9. Для каждой цифры создается «корзина» («bucket»), после чего строки  $s_i$  распределяются по «корзинам» в соответствии с  $i$ -ой цифрой с конца. Строки, у которых  $i$ -ая с конца цифра равна  $j$  попадают в  $j$ -ую корзину (например, строка 123 на первой фазе попадет в третью корзину, на второй — во вторую, на третьей — в первую). После этого элементы извлекаются из корзин в порядке увеличения номера корзины. Таким образом, после первой фазы строки отсортированы по последней цифре, после двух фаз — по двум последним, ..., после  $m$  фаз — по всем. При важно, чтобы элементы в корзинах сохраняли тот же порядок, что и в исходном массиве (до начала этой фазы). Например, если массив до первой фазы имеет вид: 111, 112, 211, 311, то элементы по корзинам распределятся следующим образом: в первой корзине будет. 111, 211, 311, а второй: 112. Напишите программу, детально показывающую работу этого алгоритма на заданном массиве.

### Формат ввода

Первая строка входного файла содержит целое число  $n$  ( $1 \leq n \leq 1000$ ). Последующие  $n$  строк содержат каждая по одной строке  $s_i$ . Длины всех  $s_i$ , одинаковы и не превосходят 20. Все  $s_i$  состоят только из цифр от 0 до 9.

### Формат вывода

В выходной файл выведите исходный массив строк  $v$ , состояние «корзин» после распределения элементов по ним для каждой фазы и отсортированный массив. Следуйте формату, приведенному в примере.

### Пример

Ввод	Вывод
9 12 32 45 67 98 29 61 35 09	Initial array: 12, 32, 45, 67, 98, 29, 61, 35, 09 *****  Phase 1 Bucket 0: empty Bucket 1: 61 Bucket 2: 12, 32 Bucket 3: empty Bucket 4: empty Bucket 5: 45, 35 Bucket 6: empty Bucket 7: 67 Bucket 8: 98 Bucket 9: 29, 09 *****  Phase 2 Bucket 0: 09 Bucket 1: 12 Bucket 2: 29 Bucket 3: 32, 35 Bucket 4: 45 Bucket 5: empty Bucket 6: 61, 67 Bucket 7: empty Bucket 8: empty Bucket 9: 98 *****  Sorted array: 09, 12, 29, 32, 35, 45, 61, 67, 98

Язык Python 3.9 (PyPy 7.3.11)

Набрать здесь Отправить файл

```
1 fin = open('input.txt')
2 N = int(fin.readline())
3 array = [fin.readline().rstrip() for _ in range(N)]
4 M = len(array[0])
5
6 print('Initial array:')
7 print(' '.join(array))
8 print('*****')
9
10 phase = 1
11 while phase <= M:
12     print(f'Phase {phase}')
13     buffer = [0]*N
14     bucket = [0]*10
15     bucket_for_print = [[] for _ in range(10)]
16     bucket_indices = [0]*10
17
18     for num in array:
19         bucket[int(num[M-phase])] += 1
20
21     for i in range(1, 10):
22         bucket_indices[i] = bucket[i-1]+bucket_indices[i-1]
23
24     for num in array:
25         buffer[bucket_indices[int(num[M-phase])]] = num
26         bucket_indices[int(num[M-phase])] += 1
27         bucket_for_print[int(num[M-phase])].append(num)
28
29     # print
30     for i in range(10):
31         if bucket_for_print[i] == []: bucket_for_print[i] = ['empty']
32
33     for i in range(10):
34         print(f'Bucket {i}: {" ".join(bucket_for_print[i])}')
35     # end of print
36
37 array[:] = buffer
38 phase += 1
```

Отправить 1 осталось 99 попыток

Предыдущая

Время посылки	ID	Задача	Компилятор	Вердикт	Тип посылки	Время	Память	Тест	Баллы
3 ноя 2023, 04:31:42	95733532	E	Python 3.9 (PyPy 7.3.11)	OK	-	234ms	28.32Mb	-	-