

# Строки, сортировка

Занятие 17 в ФМЛ 5 г Долгопрудного

# План

- Решим задачу
- Поговорим об сортировках
- Оценим асимптотики

# Тем, кто все сортировки знает

О: Подстрока-10 числа 3523014: сумма цифр = 10

Пример:

3523014 3523014 3523014 3523014

О: Если каждая цифра числа входит в подстроку-10 — то число любит 10. Например, 3523014 любит 10, но 28546 не любит 10 .

Пусть  $T(n)$  количество любящих 10 чисел от  $[1..10^n]$ . Например  $T(2) = 9$  и  $T(5) = 3492$ .

Найти  $T(10^{18}) \% 1\,000\,000\,007$ .

# Пузырьковая?

- Bubble Sort

# Реализация на python

```
def bubble(arr):  
    l = len(arr)  
    for a in range(l):  
        for b in range(l-1):  
            if (arr[a] < arr[b]):  
                arr[a], arr[b] = arr[b], arr[a]  
    return arr
```

# Вставками

- Insertion Sort

# Вставками

- Insertion Sort

```
def insertion(x):
```

```
    for i in range(1, len(x)):
```

```
        while x[i - 1] > x[i] and i >= 0:
```

```
            x[i - 1], x[i] = x[i], x[i - 1]
```

```
            i -= 1
```

```
    return x
```

# Сортировка слиянием

- MergeSort



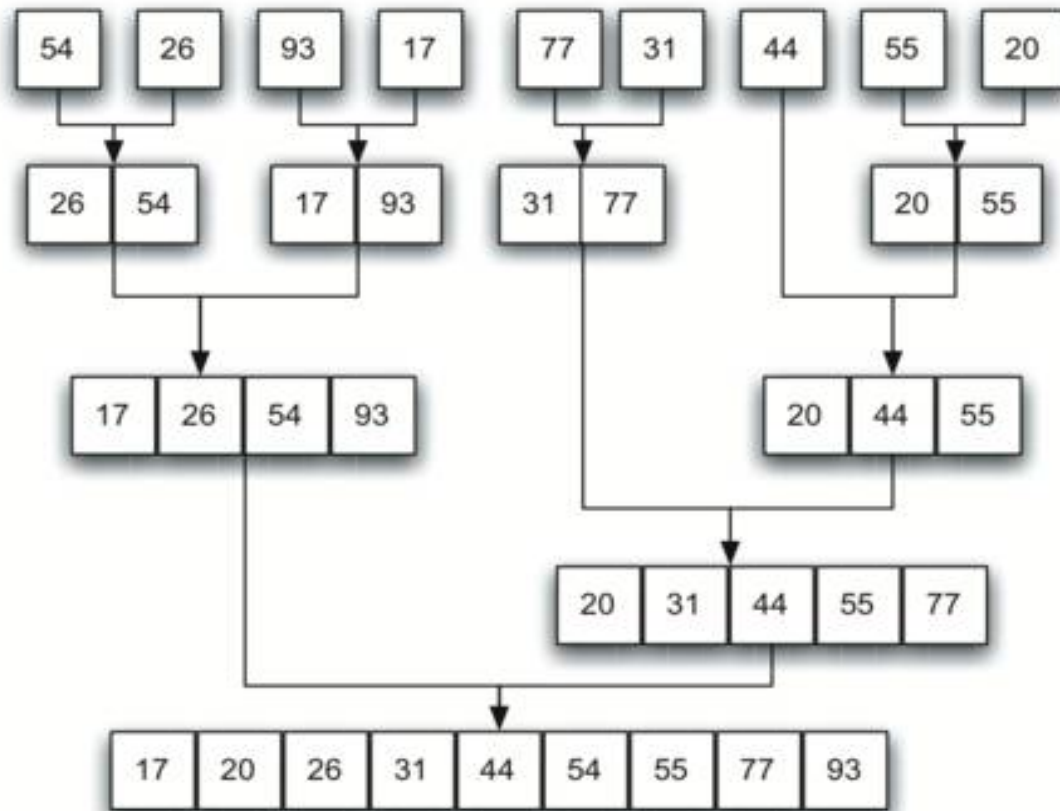
# Сортировка слиянием

```
import operator
def merge(left, right, compare):
    result=[]
    i,j=0,0
    while i<len(left) and j<len(right):
        if compare(left[i],right[j]):
            result.append(left[i])
            i+=1
        else:
            result.append(right[j])
            j+=1
    while (i<len(left)):
        result.append(left[i])
        i+=1
    while (j<len(right)):
        result.append(right[j])
        j+=1
    return result
```

```
def mergeSort(L,compare=operator.lt):
    if len(L)<2:
        return L[:]
    else:
        middle =int(len(L)/2)
        left=mergeSort(L[:middle], compare)
        right=mergeSort(L[middle:], compare)
        return merge(left,right,compare)
```

# Сортировка слиянием

- MergeSort



# Сортировка слиянием

- MergeSort

# Сортировка быстрая

- quicksort

- 



# Quicksort

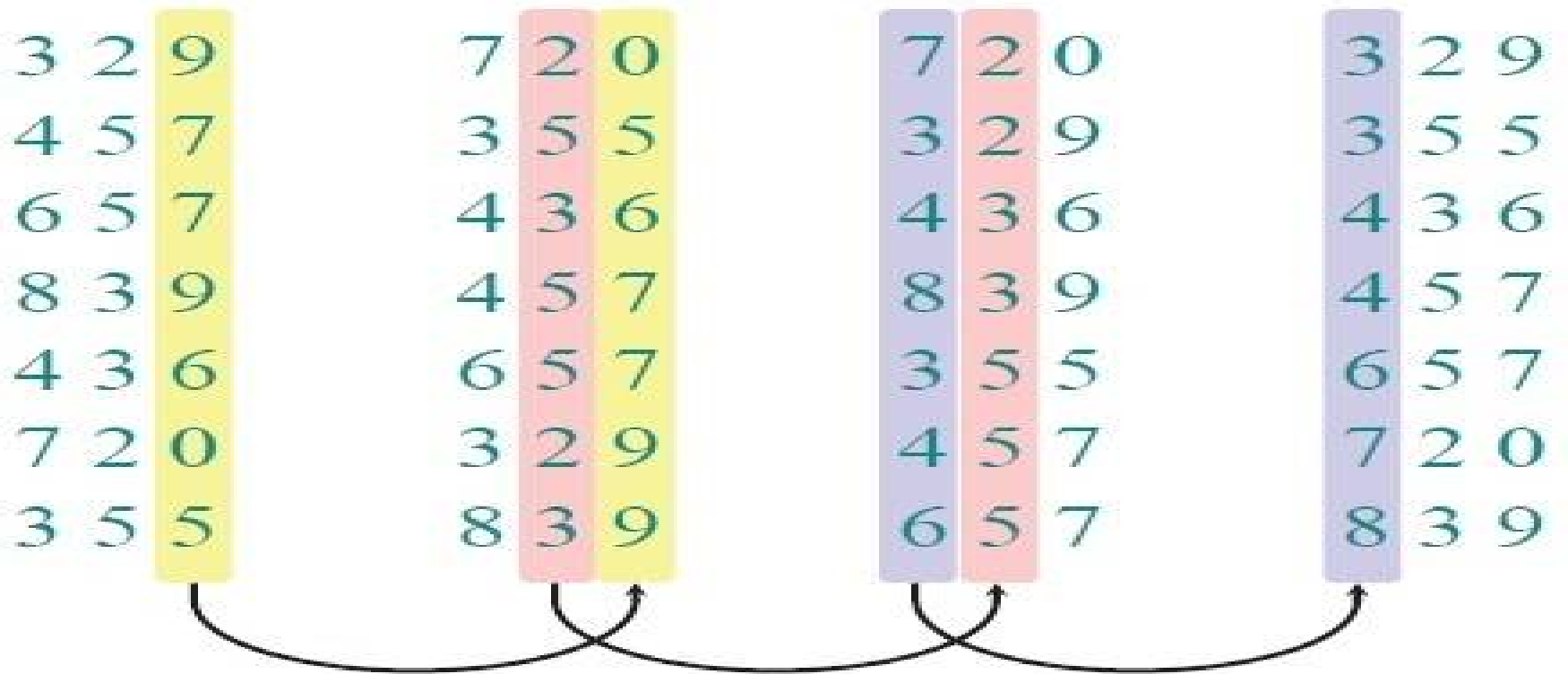
```
QUICKSORT(A,p,q)
  if(p < q)
  then r = PARTITION(A,p,q)
    QUICKSORT(A,p,r-1)
    QUICKSORT(A,r+1,q)
```

```
PARTITION(A,p,q)
  x = A[p]
  i=p
  for j = p+1 to q
    if A[j] <= x
      then i = i+1
      swap A[i] with A[j]
  swap A[p] with A[i]
  return i
```

# Radix Sort

<http://www.cppblog.com/wanghaiguang/archive/2012/11/13/195119.aspx>

## Operation of radix sort



# Bucket Sort (Карманы)

- Предположим, что необходимо отсортировать массив  $A[1...n]$ , все элементы которого равномерно распределены в интервале  $[0,1)$ . Идея заключается в том, чтобы разбить интервал на  $k$  одинаковых интервалов, или карманов (buckets).



# Карманная сортировка

```
function bucket-sort(A, n) is  
    buckets  $\leftarrow$  НОВЫЙ массив из n пустых элементов  
    for i = 0 to (length(A)-1) do  
        вставить A[i] в конец массива buckets[msbits(A[i], k)]  
    for i = 0 to n - 1 do  
        next-sort(buckets[i])  
    return Конкатенация массивов buckets[0], ..., buckets[n-1]
```

(c)[https://en.wikipedia.org/wiki/Bucket\\_sort](https://en.wikipedia.org/wiki/Bucket_sort)

# Вопросы?

- Практикум:
  - Реализация Карманной сортировки
  - Полегче — реализация поразрядной сортировки
  - посложнее — первая задача