

# Сортировки

- ♦ поразрядная сортировка (radix sort)
- ♦ пузырьковая сортировка (bubble sort)
- ♦ сортировка кучей (heap sort)
- ♦ сортировка слиянием (merge sort)

# Поразрядная сортировка

♦ Другие названия: цифровая сортировка, radix sort

- ✓ чисел
- ✓ строк
- ✓ дат
- ✓ структур с несколькими полями



♦ Класс: сортировки без сравнения, distribution sort

♦ Основная идея: последовательное (многократное) раскидывание сортируемых элементов по кучкам (buckets) на основании равенства определенного поля этих элементов и последующее объединение кучек в определенном порядке.

После того, как эта процедура будет повторена последовательно для всех полей (напр., разрядов числа), получившаяся кучка будет содержать отсортированную последовательность элементов.

03	-	января	-	2014
14	-	января	-	2013
03	-	апреля	-	2013

порядок действий:

1. отсортировать по годам
2. отсортировать по месяцам
3. отсортировать по числам



# Поразрядная сортировка

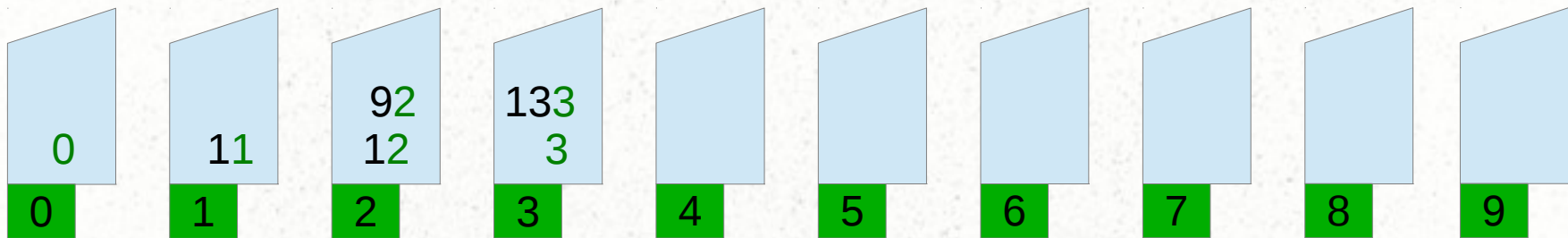
- ♦ Сортировка массива чисел: [ 12, 11, 0, 92, 3, 133 ]

Самое длинное число 133 имеет 3 разряда

Значит описанную процедуру необходимо повторить 3 раза

- ♦ Шаг 1а: разложить числа по корзинам по младшему разряду (по единицам)

[ 12, 11, 0, 92, 3, 133 ]



- ♦ Шаг 1б: обойти корзины и собрать из них числа, сохраняя порядок чисел в корзине (в каком порядке числа попали в корзину, в таком они и достаются)

[ 0, 11, 12, 92, 3, 133 ]

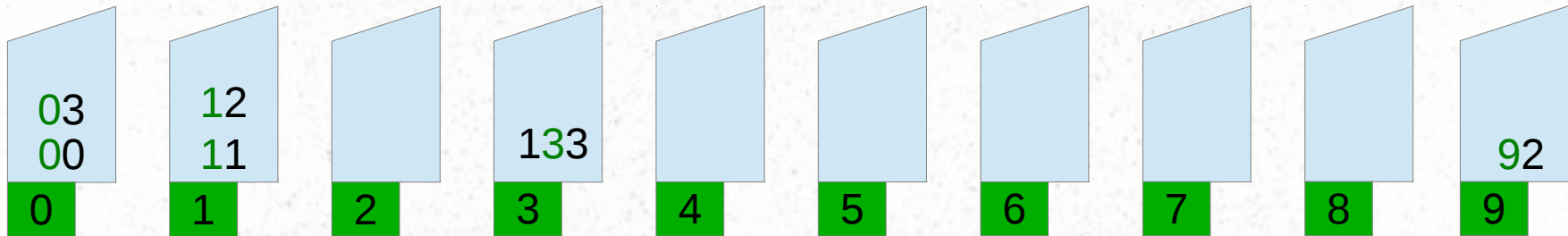
# Поразрядная сортировка

- Массив после Шага 1:

[ 0, 11, 12, 92, 3, 133 ]

- Шаг 2а: разложить числа по корзинам по второму разряду (по десяткам)

[ 00, 11, 12, 92, 03, 133 ]



- Шаг 2б: обойти корзины и собрать из них числа, сохраняя порядок чисел в корзине (в каком порядке числа попали в корзину, в таком они и достаются)

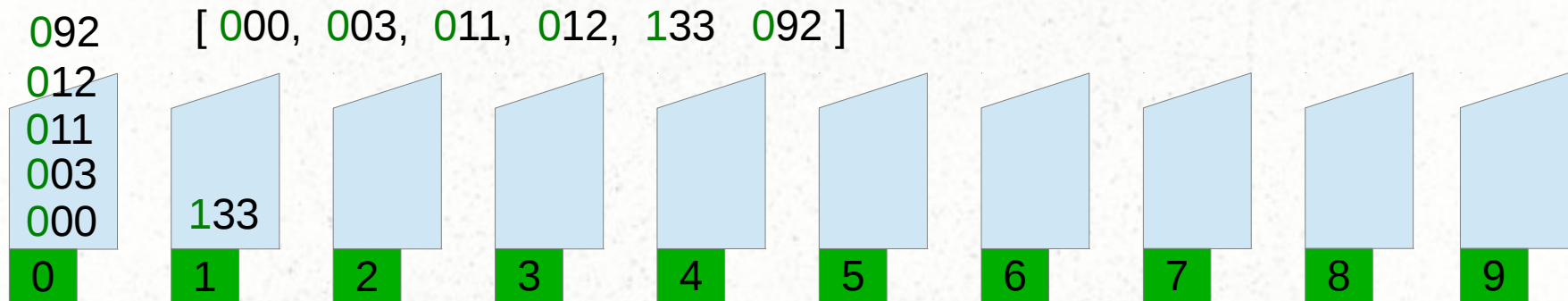
[ 00, 03, 11, 12, 133, 92 ]

# Поразрядная сортировка

- Массив после Шага 2:

[ 0, 3, 11, 12, 133, 92 ]

- Шаг 3а: разложить числа по корзинам по третьему разряду (по сотням)



- Шаг 3б: обойти корзины и собрать из них числа, сохраняя порядок чисел в корзине (в каком порядке числа попали в корзину, в таком они и достаются)

[ 000, 003, 011, 012, 092, 133 ]

- Готово! Отсортированный массив:

[ 0, 3, 11, 12, 92, 133 ]

- Вопрос: какие корзины были бы заполнены на 4-ой итерации?

Все попало бы в корзину #0



# Структура данных

- ♦ Как представить одну корзину?
- ♦ Как представить все корзины?

## ♦ Корзина (bucket)

#4	092
#3	012
#2	011
#1	003
#0	000

- ♦ должна уметь хранить множество чисел
- ♦ должна уметь сохранять порядок добавления

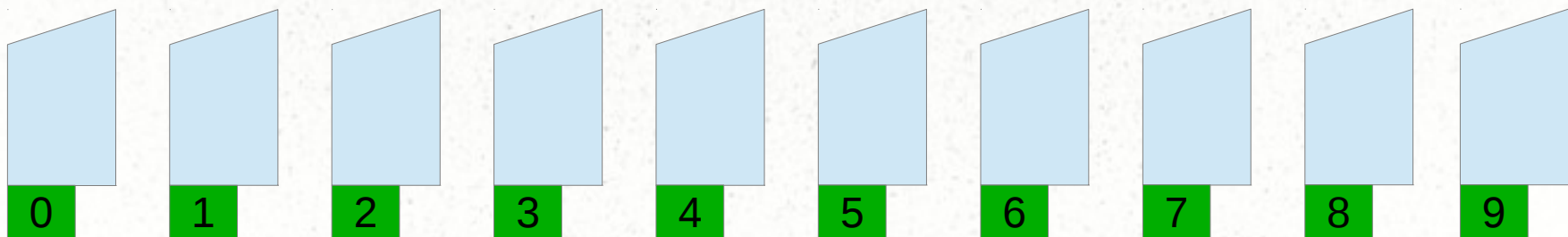


Для этих целей подходит массив!



## ♦ Все корзины:

- ♦ отдельных корзин много
- ♦ порядок корзин имеет значение



Вся система корзин: массив массивов

# Получение значения разряда

- ♦ Задача: как получить значение заданного разряда заданного числа?

12345    #=>5  
         #=>4  
         #=>3

- ♦ Способ 1: преобразовать в строку и получить нужную позицию из строки

```
num=12345  
num.to_s[4,1] #=> 5  
num.to_s[3,1] #=> 4  
num.to_s[2,1] #=> 3
```

# Получение значения разряда

- ◆ Способ 2: при помощи арифметических операций

num=12345

- ◆ Чтобы получить значение в разряде единиц (в позиции 1 с конца) = 5

12345 : 10 = 1234.5

обычное, не целочисленное, деление

12345 % 10 = 5

остаток от деления

- ◆ Чтобы получить значение в разряде десятков (в позиции 2 с конца) = 4

12345 : 100 = 123.45

обычное, не целочисленное, деление

12345 % 100 = 45

остаток от деления

45 / 10 = 4

целочисленное деление

- ◆ Чтобы получить значение в разряде сотен (в позиции 3 с конца) = 3

12345 : 1000 = 12.345

обычное, не целочисленное, деление

12345 % 1000 = 345

остаток от деления

345 / 100 = 3

целочисленное деление



# Получение значения разряда

- Можно ли сформулировать закономерность?

$n=12345$	$p=1$	$12345 \% 10 / 1 = 5$	$12345 \% 10^1 / 10^0 = 5$
$n=12345$	$p=2$	$12345 \% 100 / 10 = 4$	$12345 \% 10^2 / 10^1 = 4$
$n=12345$	$p=3$	$12345 \% 1000 / 100 = 3$	$12345 \% 10^3 / 10^2 = 3$
$n=12345$	$p=4$	$12345 \% 10000 / 1000 = 2$	$12345 \% 10^4 / 10^3 = 2$

- Можно ли вывести одну формулу? Можно ли выразить изменяемые части в арифметических выражениях через переменную  $p$  и через 10 (основание исчисления в десятиричной системе)?

$$n \% 10^p / 10^{(p-1)}$$

приоритет операции возведения в степень  $^$  выше, чем у операций  $\%$  и  $/$

$$n \% (10^p) / (10^{(p-1)})$$

# Задания (наконец!)

- ♦ Разработайте метод, возвращающий значение заданного разряда в заданном числе. Добавьте тесткейсы (см. примеры в начальном скрипте).

```
digit_at_pos(12345, 1) #=> 5
```

```
digit_at_pos(12345, 2) #=> 4
```

```
digit_at_pos(12345, 3) #=> 3
```

Начальный скрипт: `lesson.20/radix_sort/radix_sort_stub`

- ♦ Разработайте метод `radix_sort`, реализующий поразрядную сортировку массива чисел произвольной разрядности (длины). Добавьте тесткейсы.

продолжайте скрипт: `lesson.20/radix_sort/radix_sort_stub`

ответ: `lesson.20/radix_sort/radix_sort`

- ♦ (необязательное задание) Разработайте метод `radix_sort_strings`, сортирующий строки в словарном порядке

ответ: нет уж :)

# Сортировка кучей

Hello



# *Сортировка слиянием*

Hello