

# КУРС «БАЗОВЫЕ ПРИЕМЫ ПРОГРАММИРОВАНИЯ»

Часть 1. Классические алгоритмы сортировки  
линейных списков

2022 – 2023 УЧЕБНЫЙ ГОД

# Сортировка «пузырьком»

```
def sortBubble(a):  
    for j in range(1, len(a) - 1):  
        for i in range(len(a) - 1):  
            if a[i] > a[i + 1]:  
                v = a[i]  
                a[i] = a[i + 1]  
                a[i + 1] = v  
  
    return a
```

Исходный список:

[4, 6, 3, 1, 5]

Сортировка

[4, 3, 1, 5, 6]

[3, 1, 4, 5, 6]

[1, 3, 4, 5, 6]

Список после сортировки пузырьком:

[1, 3, 4, 5, 6]

# Сортировка выбором

```
def sortChoice(a):  
    for j in range(len(a)-1):  
        min_ = a[j]  
        imin=j  
        for i in range(j+1,len(a)):  
            if a[i]<min_  
                min_ = a[i]  
                imin = i  
        a[imin] = a[j]  
        a[j] = min_  
    return a
```

Исходный список:

[4, 6, 3, 1, 5]

Сортировка

[1, 6, 3, 4, 5]

[1, 3, 6, 4, 5]

[1, 3, 4, 6, 5]

[1, 3, 4, 5, 6]

Список после сортировки выбором:

[1, 3, 4, 5, 6]

# Сортировка вставкой

```
def sortInsert(a):  
    for i in range(1, len(a)):  
        v = a[i]  
        j = i-1  
        while(j >= 0 and (v < a[j])):  
            a[j+1] = a[j]  
            j = j-1  
        a[j+1] = v  
    return a
```

Исходный список:

[4, 6, 3, 1, 5]

Сортировка

[4, 6, 3, 1, 5]

[3, 4, 6, 1, 5]

[1, 3, 4, 6, 5]

[1, 3, 4, 5, 6]

Список после сортировки вставкой:

[1, 3, 4, 5, 6]

# Сортировка Шелла

```
def sortShell(a):  
    t=5  
    h=[9,5,3,2,1]  
    for j in range(t):  
        k=h[j]  
        for i in range(len(a)-k):  
            if a[i]>a[i+k]:  
                v=a[i]  
                a[i]=a[i+k]  
                a[i+k]=v  
        print(a)  
    return a
```

Исходный список:

[4, 6, 3, 1, 5]

Сортировка

[4, 6, 3, 1, 5]

[4, 6, 3, 1, 5]

[1, 5, 3, 4, 6]

[1, 4, 3, 5, 6]

[1, 3, 4, 5, 6]

Список после сортировки Шелла:

[1, 3, 4, 5, 6]

# Быстрая сортировка

```
def sortQuick(a, l, r):
    i=l
    j=r
    m=(l + r)//2
    x1=a[m]
    while i<=j:
        while a[i]<x1:
            i +=1
        while a[j]>x1:
            j -=1
        if i<=j:
            y1=a[i]
            a[i]=a[j]
            a[j]=y1
            i +=1
            j -=1
    if l<j:
        sortQuick(a, l, j);
    if i<r:
        sortQuick(a, i, r);
    return a
```

Исходный список:

[8, 9, 14, 20, 12, 7, 17, 1, 4, 11]

Сортировка

[8, 9, 11, 20, 12, 7, 17, 1, 4, 14]

[8, 9, 11, 4, 12, 7, 17, 1, 20, 14]

[8, 9, 11, 4, 1, 7, 17, 12, 20, 14]

[8, 9, 11, 4, 1, 7, 17, 12, 20, 14]

[8, 9, 7, 4, 1, 11, 17, 12, 20, 14]

[8, 9, 7, 4, 1, 11, 17, 12, 20, 14]

[1, 9, 7, 4, 8, 11, 17, 12, 20, 14]

[1, 4, 7, 9, 8, 11, 17, 12, 20, 14]

[1, 4, 7, 9, 8, 11, 17, 12, 20, 14]

[1, 4, 7, 9, 8, 11, 17, 12, 20, 14]

[1, 4, 7, 8, 9, 11, 17, 12, 20, 14]

[1, 4, 7, 8, 9, 11, 12, 17, 20, 14]

[1, 4, 7, 8, 9, 11, 12, 17, 14, 20]

[1, 4, 7, 8, 9, 11, 12, 14, 17, 20]

Список после быстрой

сортировки:

[1, 4, 7, 8, 9, 11, 12, 14, 17, 20]

Исходный список:

[4, 6, 3, 1, 5]

Сортировка

[1, 6, 3, 4, 5]

[1, 3, 6, 4, 5]

[1, 3, 6, 4, 5]

[1, 3, 4, 6, 5]

[1, 3, 4, 5, 6]

Список после

быстрой сортировки:

[1, 3, 4, 5, 6]

# КУРС «БАЗОВЫЕ ПРИЕМЫ ПРОГРАММИРОВАНИЯ»

Часть 2. Неклассические алгоритмы сортировки  
линейных списков

2022 – 2023 УЧЕБНЫЙ ГОД

# Нестандартная сортировка

Типы нестандартной сортировки в лабораторной работе № 2:

1. Отсортировать список по возрастанию младших цифр элементов списка
2. Отсортировать список по возрастанию старших цифр элементов списка

Выделение младшей цифры  $m$  в числе:

```
m = x % 10
```

Выделение старшей цифры в числе:

```
while x > 9:  
    x = x // 10
```

Функция, возвращающая старшую цифру числа:

```
def highDigit(x):  
    while x > 9:  
        x = x // 10  
    return x
```



# Сортировка «пузырьком»

по возрастанию младших цифр элементов списка

```
def sortBubbleLow(a):  
    for j in range(1, len(a) - 1):  
        for i in range(len(a) - 1):  
            if a[i] % 10 > a[i + 1] % 10:  
                v = a[i]  
                a[i] = a[i + 1]  
                a[i + 1] = v  
  
    return a
```

Исходный список:

[28, 47, 59, 10, 63, 28, 56, 37, 33, 16]

Сортировка

[47, 28, 10, 63, 28, 56, 37, 33, 16, 59]

[47, 10, 63, 28, 56, 37, 33, 16, 28, 59]

[10, 63, 47, 56, 37, 33, 16, 28, 28, 59]

[10, 63, 56, 47, 33, 16, 37, 28, 28, 59]

[10, 63, 56, 33, 16, 47, 37, 28, 28, 59]

[10, 63, 33, 56, 16, 47, 37, 28, 28, 59]

[10, 63, 33, 56, 16, 47, 37, 28, 28, 59]

[10, 63, 33, 56, 16, 47, 37, 28, 28, 59]

[10, 63, 33, 56, 16, 47, 37, 28, 28, 59]

Список после сортировки пузырьком:

[10, 63, 33, 56, 16, 47, 37, 28, 28, 59]

# Сортировка «пузырьком»

по возрастанию старших цифр элементов списка

```
def highDigit(x):  
    while x>9:  
        x = x//10  
    return x  
  
def sortBubbleHigh(a):  
    for j in range(1,len(a)-1):  
        for i in range(len(a)-1):  
            if highDigit(a[i])> highDigit(a[i+1]):  
                v = a[i]  
                a[i] = a[i+1]  
                a[i+1] = v  
  
    return a
```

Исходный список:

[918, 868, 114, 772, 767, 463]

Сортировка

[868, 114, 772, 767, 463, 918]

[114, 772, 767, 463, 868, 918]

[114, 772, 463, 767, 868, 918]

[114, 463, 772, 767, 868, 918]

[114, 463, 772, 767, 868, 918]

Список после сортировки пузырьком

:

[114, 463, 772, 767, 868, 918]

# Лабораторная работа № 1

## Часть 2. Пример программы

2022 – 2023 УЧЕБНЫЙ ГОД

# Пример лабораторной работы № 2

```
# Раздел импорта модулей
import random
```

```
# Задание
```

```
def task():
    print("Лабораторная работа № 2")
    print("Вариант № 1. Выполнил студент группы 6101-090301D Иванов П.С.")
    print("Задание:")
    print("1. В списке целочисленных элементов найти максимальный")
    print("    нечетный двузначный элемент")
    print("2. С использованием цикла while найти в списке индекс")
    print("    последнего четного элемента, кратного заданному числу")
    print("3. Отсортировать список (без использования стандартных")
    print("    функций сортировки) по возрастанию старших цифр элементов")
    print("    списка (сортировка выбором)")
    print("")
```

# Пример лабораторной работы № 2

```
# Ввод элементов списка
def inputList(a):
    a = list(map(int, input().split()))
    return(a)

# Формирование списка случайных элементов
def randomList(a):
    b, c = map(int, input().split())
    for i in range(n):
        a.append(random.randint(b, c))
    return(a)
```

# Пример лабораторной работы № 2

```
# Поиск максимального нечетного двузначного элемента
def findMax(a):
    m = -10**10
    for i in range( len(a) ):
        if (a[i]%2 !=0) and (a[i]>9) and (a[i]<100) and (a[i]>m):
            m = a[i]
    return m

# Поиск индекса последнего четного элемента,
# кратного заданному числу
def lastEl(a, x):
    i = len(a) - 1
    while (i>-1) and ((a[i]%2 != 0) or (a[i] % x != 0)):
        i -=1
    return i
```

# Пример лабораторной работы № 2

```
# Старшая цифра числа
def highDigit(x):
    while x>9:
        x = x//10
    return x
```

# Пример лабораторной работы № 2

```
# Сортировка выбором по возрастанию старших цифр элементов
def sortChoice(a):
    for j in range(len(a)-1):
        min_ = a[j]
        imin = j
        for i in range(j+1, len(a)):
            if highDigit(a[i]) < highDigit(min_):
                min_ = a[i]
                imin = i
        a[imin] = a[j]
        a[j] = min_
    return a
```



# Пример лабораторной работы № 2

```
# Вызовы функций
```

```
task()
```

```
print("Введите способ заполнения списка:")
```

```
print("1 - ввод элементов списка в одну строку через пробел:")
```

```
print("любое число-авт. формирование списка из n элементов:")
```

```
v = int(input())
```

```
print("")
```

# Пример лабораторной работы № 2

```
a=[]
if v ==1:
print("Введите в строку элементы списка:")
    a = inputList(a)
else:
    n = int(input("Введите количество элементов списка: "))
    print("Введите диапазон элементов:")
    a = randomList(a)
    print(a)
print("")
```

# Пример лабораторной работы № 2

```
m = findMax(a)
if m != -10**10:
    print("Максимальный нечетный двузначный элемент = ", m)
else:
    print("Максимальный нечетный двузначный элемент отсутствует")
print("")

x = int(input("Введите число, кратность которому нужно проверить:"))
ind = lastEl(a, x)
if ind > -1:
    print("Индекс последнего четного элемента, кратного заданному числу = ", ind)
else:
    print("В списке отсутствуют четные элементы, кратные заданному числу")
print("")
```

# Пример лабораторной работы № 2

```
print("Исходный список:")  
print(a)  
sortChoice(a)  
print("Список после сортировки выбором:")  
print(a)
```

# Примеры экрана выполнения программы

Лабораторная работа № 2

Вариант № 1. Выполнил студент группы 6101-090301D Иванов П.С.

Задание:

1. В списке целочисленных элементов найти максимальный нечетный двузначный элемент
2. С использованием цикла while найти в списке индекс последнего четного элемента, кратного заданному числу
3. Отсортировать список (без использования стандартных функций сортировки) по возрастанию старших цифр элементов списка (сортировка выбором)

Введите способ заполнения списка:

1 - ввод элементов списка в одну строку через пробел:  
любое число - автоматическое формирование списка из n элементов:  
1

Введите в строку элементы списка:

44 16 2 5 21 36 12 58

Максимальный нечетный двузначный элемент = 21

Введите число, кратность которому нужно проверить: 3

Индекс последнего четного элемента, кратного заданному числу = 6

Исходный список:

[44, 16, 2, 5, 21, 36, 12, 58]

Список после сортировки выбором по возрастанию старших цифр:

[16, 12, 2, 21, 36, 44, 5, 58]

Лабораторная работа № 2

Вариант № 1. Выполнил студент группы 6101-090301D Иванов П.С.

Задание:

1. В списке целочисленных элементов найти максимальный нечетный двузначный элемент
2. С использованием цикла while найти в списке индекс последнего четного элемента, кратного заданному числу
3. Отсортировать список (без использования стандартных функций сортировки) по возрастанию старших цифр элементов списка (сортировка выбором)

Введите способ заполнения списка:

1 - ввод элементов списка в одну строку через пробел:  
любое число - автоматическое формирование списка из n элементов:  
2

Введите количество элементов списка: 10

Введите диапазон элементов:

10 500

[220, 330, 305, 308, 110, 240, 489, 223, 395, 458]

Максимальный нечетный двузначный элемент отсутствует

Введите число, кратность которому нужно проверить: 19

В списке отсутствуют четные элементы, кратные заданному числу

Исходный список:

[220, 330, 305, 308, 110, 240, 489, 223, 395, 458]

Список после сортировки выбором по возрастанию старших цифр:

[110, 220, 240, 223, 330, 305, 308, 395, 489, 458]