АКАДЕМИЯ АЙТИ

День 9

academyit.ru

Типовые задачи на обработку текста.

Практические примеры составления блоксхем и псевдокода.
Простейшие алгоритмические задачи.
Перевод алгоритма в код.
Подпрограммы (функции) как основные блоки кода.

Турашова Анна Николаевна
Преподаватель
anna1turashova@gmail.com
Telegram: @anna1tur

Поверка домашнего задания

Задача 1

Отсортировать слова в тексте по алфавиту, сохраняя регистр букв, но удаляя знаки препинания.

Ввод:

Добрый день, дорогие друзья! Сегодня ХОРОШАЯ погода!

Вывод:

день Добрый дорогие друзья погода Сегодня ХОРОШАЯ

Задача 2

Необходимо отредактировать текст следующим образом:

1. Заменить неприличные слова на ***
2. Первая буква в предложении должна начинаться с заглавной буквы, а остальные – с маленькой.

Ввод:

Добрый мат день, дорогие блин друзья. сегодня ХОРОШАЯ мат погода!

Вывод:

Добрый *** день, дорогие *** друзья. Сегодня хорошая *** погода!

Повторение:
методы списков, кортежей

Методы списков

- Многие **методы списков**, в отличие от методов строк **изменяют сам список**, а потому результат выполнения не нужно записывать в отдельную переменную

Методы поиска значений

Метод	Назначение	Пример
a.count(x)	Количество вхождений элемента в списке, если нет совпадений - 0	a = [1, 2, 3, 2, 3, 1, 2] print(a.count(2)) Результат: 3
a.index(x)	Найти индекс первого вхождения конкретного элемента	a =[12,34,23,45] print(a.index(23)) Результат: 2
x in a	Проверка, что x содержится в a	5 in [2, 3, 5]
x not in a	Проверка, что x не содержится в a То же, что и not (x in a)	5 not in [2, 3, 6]
max(a)	Максимальный элемент списка	max([2, 3, 7]) == 7
min(a)	Минимальный элемент списка	min([2, 3, 7]) == 2
sum(a)	Сумма элементов списка	sum([2, 3, 7]) == 12

Добавление в список

Метод	Назначение	Пример
a.append(x)	Добавить x в конец a	a = [2, 3, 7] a.append(8) a == [2, 3, 7, 8]
a.extend(a2)	Добавить элементы последовательности a2 в конец a	a = [2, 3, 7] a.extend([8, 4, 5]) a == [2, 3, 7, 8, 4, 5]
a.insert(n, x)	Вставить x в a на позицию n, подвинув последующую часть дальше	a = [2, 3, 7] a.insert(0, 8) a == [8, 2, 3, 7]

Удаление из списка

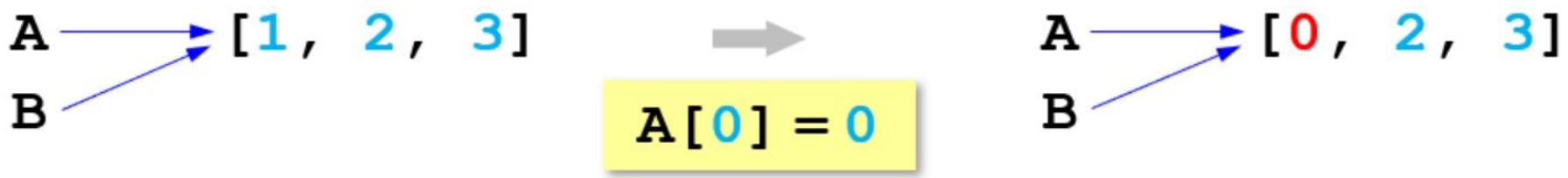
Метод	Назначение	Пример
del a[n]	Удалить n-й элемент списка	a = [2, 3, 7] del a[1] a == [2, 7]
del a[start:stop:step]	Удалить из a все элементы, попавшие в срез	a = [2, 3, 7] del a[:2] a == [7]
a.clear()	Удалить из a все элементы (то же, что del a[:])	a.clear()
a.remove(x)	Удалить первое вхождение x в a, в случае x not in a — ошибка	a = [2, 3, 7] a.remove(3) a == [2, 7]
a.pop(n)	Получить n-й элемент списка и одновременно удалить его из списка. Без аргументов - последний элемент a.pop() == a.pop(-1)	a = [2, 3, 7] a.pop(1) == 3 a == [2, 7]

Преобразования списков

Метод	Назначение	Пример
a.reverse()	Изменить порядок элементов в a на обратный	a = [2, 3, 7] a.reverse() a == [7, 3, 2]
a.sort()	Отсортировать список по возрастанию	a = [3, 2, 7] a.sort() a == [2, 3, 7]
a.sort(reverse=**True**)	Отсортировать список по убыванию	a = [3, 2, 7] a.sort(reverse = True) a == [7, 3, 2]
sorted(a)	Возвращает отсортированный список	a = [3, 2, 7] print(sorted(a)) [2, 3, 7]

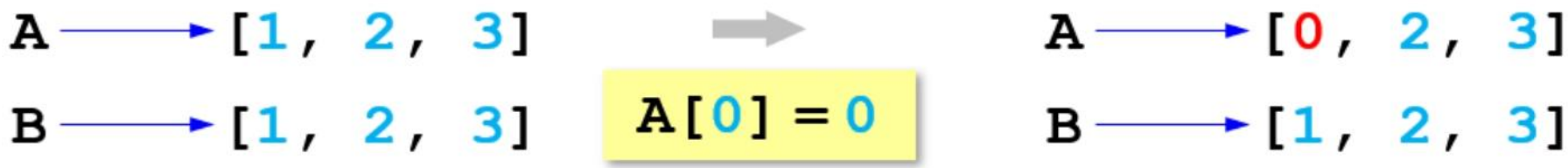
Копирование списка

```
A = [1, 2, 3]
B = A
```




```
A = [1, 2, 3]
B = A[:]
```

копия массива A



a.copy() – метод для копирования

Преобразование списка в строку

Метод строки join принимает один аргумент — список слов, которые нужно склеить.

Разделителем (точнее, «соединителем») служит та самая строка, чей метод join вызывается.

```
s = ['Вышел', 'Зайчик', 'Погулять']
print(''.join(s)) 'ВышелЗайчикПогулять'
print(' '.join(s)) 'Вышел Зайчик Погулять'
print('-'.join(s)) 'Вышел-Зайчик-Погулять'
print('! '.join(s))'Вышел! Зайчик! Погулять'
```

Кортежи

Кортежи в Python — это неизменяемые последовательности элементов.

Они полезны в тех случаях, когда необходимо передать данные, не позволяя изменять их.

Кортеж – это структура данных, которая используется для хранения последовательности упорядоченных и неизменяемых элементов.

Особенности кортежей

1) Кортеж создается со значениями, разделенными запятой.
2) В кортеже нельзя добавлять или удалять его элементы.
3) Можно создавать вложенные кортежи.
4) Можно получить доступ к элементам кортежа через их индекс, сделать срезы
5) Поддерживают две операции с кортежами:
   + для объединения
   * для повторения элементов.
7) Можно использовать in и not in.
8) Можно перебирать элементы с помощью цикла for.
9) Класс кортежей Python имеет две функции — count() и index() .

Создание кортежа

- Пустой кортеж

```
empty_tuple =()
```

- Заполненный элементами

```
tuple_numbers = (1, 2, 3, 1)
tup = 1, 2, 3, 1, None, "Hello"
one = (1, )
cake = tuple('cake')
print(type(cake))
```

Результат: <class 'tuple'>

- Вложенный кортеж

```
nested_tuple = ((1, 2), ("Hello", "Hi"), "Python")
```

Сравнение и присваивание кортежей

```
(1, 2) == (1, 3) # False
(1, 2) < (1, 3) # True
(1, 2) < (5,) # True
('7', 'червей') < ('7', 'треф') # False
# А вот так сравнивать нельзя:
элементы кортежей разных типов
(1, 2) < ('7', 'пик')
```

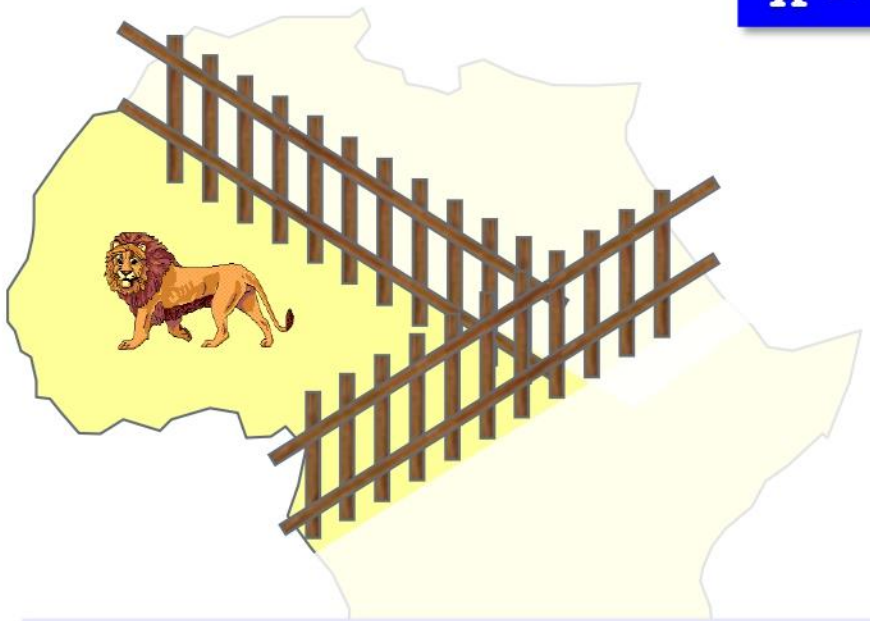
Хранение элементов в неупорядоченном списке

Поиск элемента (in, index)	O(n)
Добавление элемента (append)	O(1)
Удаление элемента (del)	O(n)
Поиск минимального/максимального элемента	O(n)

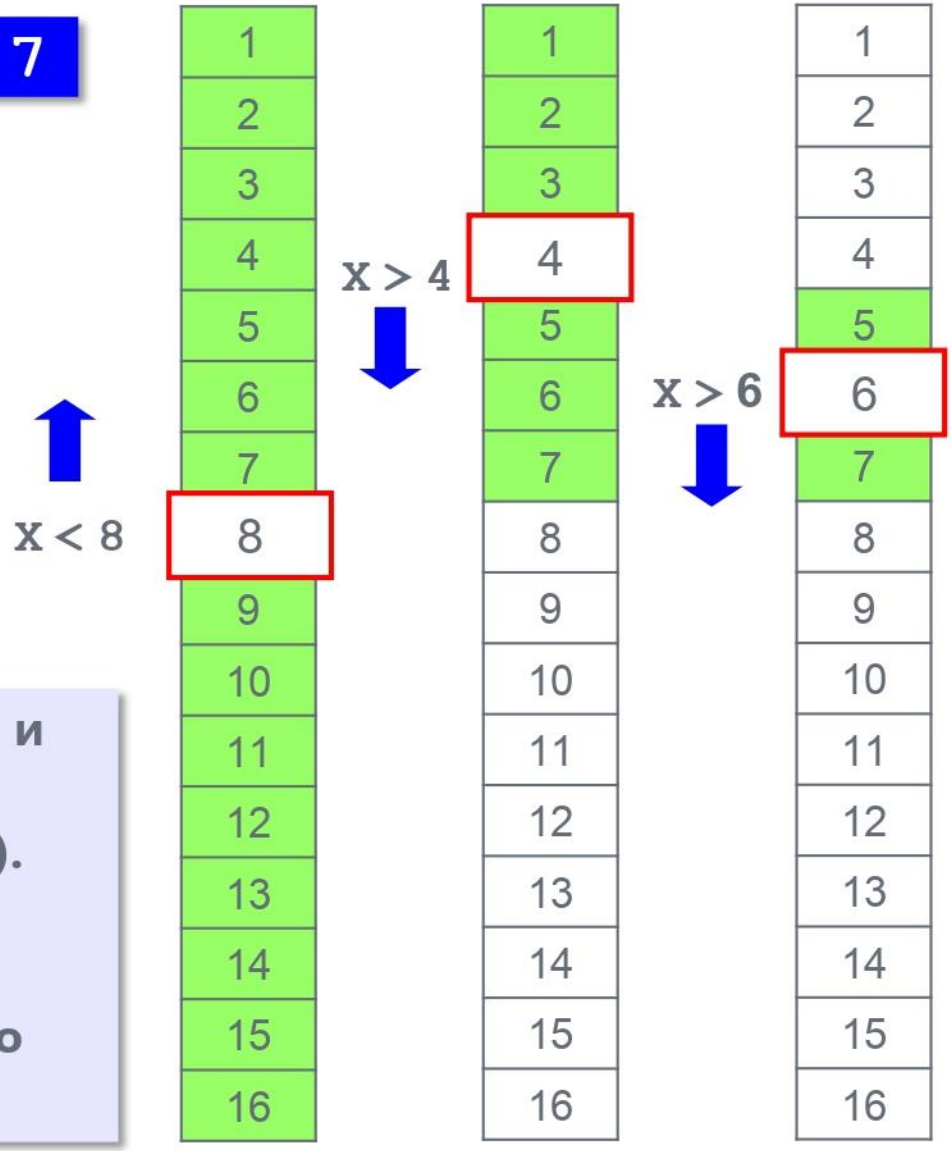
Хранение элементов в упорядоченном списке

Поиск элемента (двоичный поиск)	O(log n)
Добавление элемента (вставка)	O(n)
Удаление элемента (del)	O(n)
Поиск минимального/максимального элемента	O(1)

Двоичный поиск



X = 7



1. Выбрать средний элемент `A[c]` и сравнить с X.
2. Если `X == A[c]`, то нашли (**стоп**).
3. Если `X < A[c]`, искать дальше в первой половине.
4. Если `X > A[c]`, искать дальше во второй половине.

Вложенные списки. Генераторы списков.

Вложенные списки

Вложенные списки — это списки, которые включают списки как элементы.

В Python нет ограничений в уровнях вложенности, но для практических целей наиболее полезны двумерные вложенные списки – матрицы.

Матрица — это прямоугольная таблица, составленная из элементов одного типа (чисел, строк и т.д.). Каждый элемент матрицы имеет два индекса – номера строки и столбца.

Создание матриц

Матрица – это список списков

```
a = [[-1, 0, 1],
     [-1, 0, 1],
     [0, 1, -1]]
```

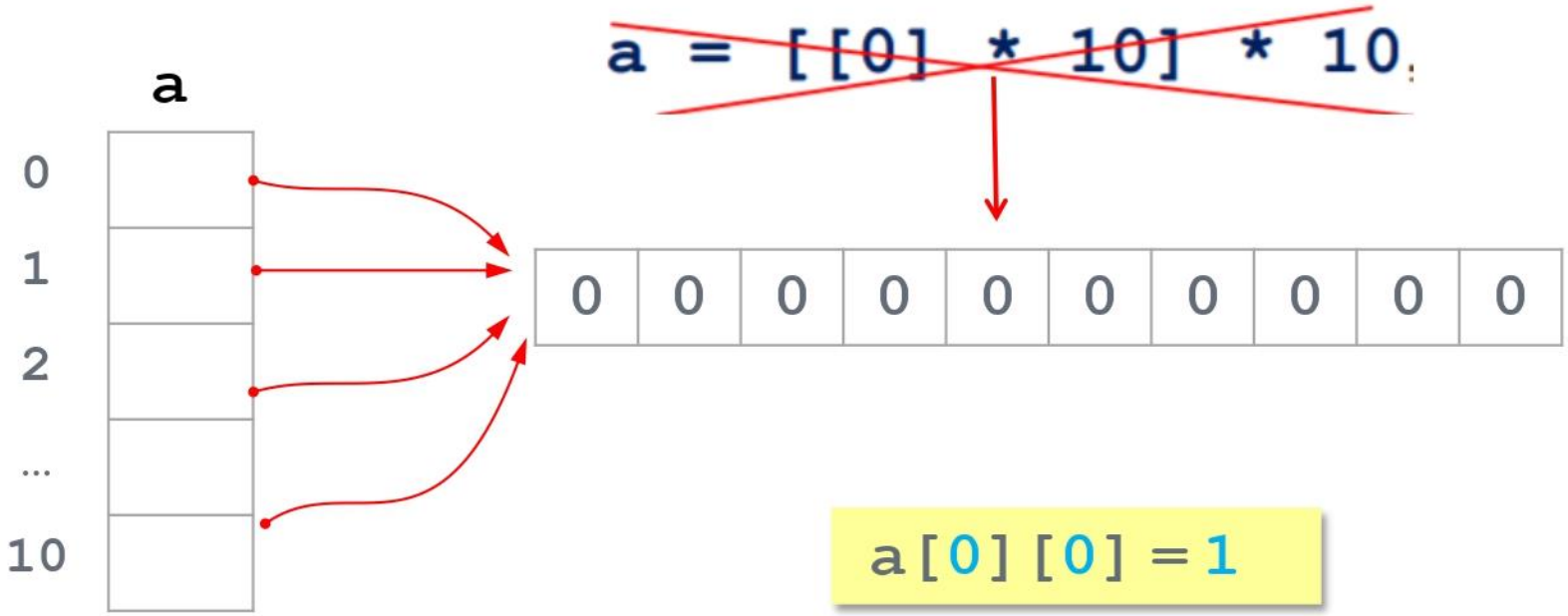
```
a = [[-1, 0, 1], [-1, 0, 1], [0, 1, -1]]
```

Каждый элемент имеет два индекса, нумерация элементов с нуля

a[0][0]	a[0][1]	a[0][2]
a[1][0]	a[1][1]	a[1][2]
a[2][0]	a[2][1]	a[2][2]

Создание нулевой матрицы

Создать нулевую матрицу размером 10 x 10 элементов.



Операция умножения будет создавать множество ссылок на одну строку – неправильно
Правильно – использовать цикл или генератор списка

```
a = [[0] * 10 for __ in range(10)]
```

Ввод элементов с клавиатуры

Каждая строка таблицы на отдельной строке, значения в строке разделяются пробелами)

```
table = []
for i in range(n):
   row = [int(x) for x in input().split()]
   table.append(row)

table = [[int(x) for x in input().split()]
     for i in range(n)]
```

Вывод элементов в консоль

```
for row in a:
    for x in row:
      print (x, end = "\t" )
    print()


for i in range(len(a)):
    for j in range(len(a[i])):
      print (f"{a[i][j]:4d}", end = "")
    print()
```

Заполнение матрицы случайными числами

```
import random
for i in range(n):
  for j in range(m):
    a[i][j] = random.randint(20, 80)
```

Перебор элементов матрицы

Перебор по строкам (вычисление суммы элементов)

```
s = 0
for i in range(n):
  for j in range(m):
    s += a[i][j]
```

```
s = 0
for row in a:
  s += sum(row)
print(s)
```

Перебор элементов матрицы

Перебор по столбцам (вычисление максимальной суммы по столбцам)

```
max_s = 0
for j in range(m):
    s = 0
    for i in range(n):
        s += a[i][j]
    max_s = max(s, max_s)
```

Выделение строк, столбцов

Выделение первой строки

```
r = a[1][:]
```

Выделение третьего столбца

```
c = [row[3] for row in a]
```

Выделение главной диагонали

```
d = [a[i][i] for i in range(len(a))]
```

Домашнее задание:

Задача №3752. Встречалось ли число раньше

Во входной строке записана последовательность чисел через пробел. Для каждого числа выведите слово `YES` (в отдельной строке), если это число ранее встречалось в последовательности или `NO`, если не встречалось.

Входные данные

Вводится список чисел. Все числа списка находятся на одной строке.

Выходные данные

Выведите ответ на задачу.

Примеры

входные данные
1 2 3 2 3 4
выходные данные
NO NO NO YES YES NO

Задача №112364. Поиск в матрице

Напишите программу, которая определяет, сколько раз встречается в матрице элемент, равный K.

Входные данные

В первой строке записаны через пробел размеры матрицы: количество строк N и количество столбцов M ($1 \le N, M \le 100$). В следующих N строках записаны строки матрицы, в каждой – по M натуральных чисел, разделённых пробелами. В следующей строке записано целое число K.

Выходные данные

Программа должна вывести количество элементов матрицы, равных K.

Примеры

входные данные
4 5 1 2 3 4 5 6 12 8 9 10 11 12 12 14 15 16 17 18 12 20 12
выходные данные
4

Задача №112368. Столбцы с максимумом

Напишите программу, которая находит в матрице столбцы, в которых есть элемент, равный максимальному.

Входные данные

В первой строке записаны через пробел размеры матрицы: количество строк N и количество столбцов M ($1 \le N, M \le 100$). В следующих N строках записаны строки матрицы, в каждой – по M натуральных чисел, разделённых пробелами.

Выходные данные

Программа должна вывести все столбцы, в которых есть элемент, равный максимальному элементу в матрице. Каждый столбец выводится в одну строку, элементы разделяются пробелами.

Примеры

входные данные
4 5 1 897 2 54 234 75 12 3 46 9 13 26 56 9 12 14 90 897 6 34
выходные данные
897 12 26 90 2 3 56 897

Задача №354. Побочная диагональ

Дано число `n`, `n \(\leq\) 100` . Создайте массив `n×n` и заполните его по следующему правилу:
- числа на диагонали, идущей из правого верхнего в левый нижний угол, равны 1;
- числа, стоящие выше этой диагонали, равны 0;
- числа, стоящие ниже этой диагонали, равны 2.

Входные данные

Программа получает на вход число `n`.

Выходные данные

Необходимо вывести полученный массив. Числа разделяйте одним пробелом.

Примеры

входные данные
4
выходные данные
0 0 0 1 0 0 1 2 0 1 2 2 1 2 2 2

Задача №355. Симметричная ли матрица?

Проверьте, является ли двумерный массив симметричным относительно главной диагонали. Главная диагональ — та, которая идёт из левого верхнего угла двумерного массива в правый нижний.

Входные данные

Программа получает на вход число `n \(\leq\) 100`, являющееся числом строк и столбцов в массиве. Далее во входном потоке идет `n` строк по `n` чисел, являющихся элементами массива.

Выходные данные

Программа должна выводить слово `yes` для симметричного массива и слово `no` для несимметричного.

Примеры

входные данные
3 0 1 2 1 5 3 2 3 4
выходные данные
yes

АПЛАНА
ГРУППА КОМПАНИЙ

Входит в ГК Аплана

АКАДЕМИЯ АЙТИ

Основана в 1995 г.

E-learning и очное обучение

Направления обучения:

- Информационные технологии
- Информационная безопасность
- ИТ-менеджмент и управление проектами
- Разработка и тестирование ПО
- Гос. и муниципальное управление

Филиалы:

Санкт-Петербург, Казань, Уфа, Челябинск, Хабаровск, Красноярск, Тюмень, Нижний Новгород, Краснодар, Волгоград, Ростов-на-Дону

Головной офис в Москве

Разработка программного обеспечения и информационных систем

Ежегодные награды Microsoft, Huawei, Cisco и другие

Программы по импортозамещению

Ресурсы более 400 высококлассных экспертов и преподавателей

Сеть региональных учебных центров по всей России

Крупные заказчики

СБЕР

РЖД

Билайн

TATNEFT

НОРНИКЕЛЬ

100+
сотрудников

АКАДЕМИЯ АЙТИ

Спасибо за внимание!

Центральный офис:
Москва, Варшавское шоссе 47, корп. 4, 7 этаж
Тел: +7 (495) 150-96-00
academy@it.ru
academyit.ru