

## Пример обработки двумерного массива

Пусть дан квадратный массив из  $n$  строк и  $n$  столбцов. Необходимо элементам, находящимся на главной диагонали, проходящей из левого верхнего угла в правый нижний (то есть тем элементам  $A[i][j]$ , для которых  $i=j$ ) присвоить значение 1, элементам, находящимся выше главной диагонали – значение 0, элементам, находящимся ниже главной диагонали – значение 2. То есть получить такой массив (пример для  $n=4$ ):

```
1 0 0 0
2 1 0 0
2 2 1 0
2 2 2 1
```

Рассмотрим несколько способов решения этой задачи.

Элементы, которые лежат выше главной диагонали – это элементы  $A[i][j]$ , для которых  $i < j$ , а для элементов ниже главной диагонали  $i > j$ . Таким образом, мы можем сравнивать значения  $i$  и  $j$  и по ним определять значение  $A[i][j]$ . Получаем следующий алгоритм:

```
for i in range(n):
    for j in range(n):
        if i < j:
            A[i][j] = 0
        elif i > j:
            A[i][j] = 2
        else:
            A[i][j] = 1
```

Данный алгоритм плох, поскольку выполняет одну или две инструкции `if` для обработки каждого элемента. Если мы усложним алгоритм, то мы сможем обойтись вообще без условных инструкций.

Сначала заполним главную диагональ, для чего нам понадобится один цикл:

```
for i in range(n):
    A[i][i] = 1
```

Затем заполним значением 0 все элементы выше главной диагонали, для чего нам понадобится в каждой из строк с номером  $i$  присвоить значение элементам  $A[i][j]$  для  $j = i + 1, \dots, n - 1$ . Здесь нам понадобятся вложенные циклы:

```
for i in range(n):
    for j in range(i + 1, n):
        A[i][j] = 0
```

Аналогично присваиваем значение 2 элементам  $A[i][j]$  для  $j = 0, \dots, i - 1$ :

```
for i in range(n):
    for j in range(0, i):
        A[i][j] = 2
```

Можно также внешние циклы объединить в один и получить еще одно, более компактное решение:

```
for i in range(n):
    for j in range(0, i):
        A[i][j] = 2
    A[i][i] = 1
    for j in range(i + 1, n):
        A[i][j] = 0
```

А вот такое решение использует операцию повторения списков для построения очередной строки списка.  $i$ -я строка списка состоит из  $i$  чисел 2, затем идет одно число 1, затем идет  $n - i - 1$  число 0:

```
for i in range(n):
    A[i] = [2] * i + [1] + [0] * (n - i - 1)
```

А можно заменить цикл на генератор:

```
A = [[2] * i + [1] + [0] * (n - i - 1) for i in range(n)]
```