

А. Диагональное произведение. 2021

Ограничение времени	0.4 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Написать функцию `get_nonzero_diag_product(X)`, которая подсчитывает произведение ненулевых элементов на диагонали прямоугольной матрицы. Если все элементы на диагонали нулевые, функция должна вернуть `None`.

Формат ввода

```
get_nonzero_diag_product(np.array([[1, 0], [0, 0]]))
```

Формат вывода

1

Примечания

Следует отправить файл `get_nonzero_diag_product.py` с реализованной функцией `get_nonzero_diag_product`. Функция должна быть написана с использованием библиотеки `Numpy`. Использование циклов категорически запрещено!

В. Вычисление матожидания. 2021

Ограничение времени	0.5 секунд
Ограничение памяти	150Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Охотники за сокровищами отправились за очередным кладом в необычное место, которое называется "Поле чудес". "Поле чудес" имеет размер H на W метров. На каждом квадратном метре "Поля чудес" закопан сундук с определённым количеством камней. Скоро "Поле чудес" должен посетить лепрекон. Под влиянием лепрекона камни в сундуках превращаются в золотые монеты. Лепрекон появляется в определённой клетке поля и действует на прямоугольную область размером h на w , верхний левый край этой области - точка, где стоит лепрекон. Охотники за сокровищами не знают, когда точно появятся лепрекон, но хотят получить как можно больше золота. Известно вероятностное распределение на появление лепрекона в каждый момент времени в каждой клетке поля. Необходимо для каждой клетки "Поля чудес" посчитать математическое ожидание награды, которое будет получено от этой клетки.

Необходимо написать функцию `calc_expectations(h, w, X, Q)`, где:

- * h, w — размеры области влияния лепрекона;
- * X — двумерная матрица целых чисел размера H на W , причем $X[i, j]$ — количество камней в клетке с координатами (i, j) ;
- * Q — двумерная матрица вещественных чисел размера H на W , причем $Q[i, j]$ — вероятность появления лепрекона в точке с координатами (i, j) .

Ограничение на размеры поля:

$$* 1 \leq H * W \leq 10^6$$

Формат ввода

```
calc_expectations(2, 2, np.array([ [1, 1, 1, 1], [2, 2, 2, 2], [3, 3, 3, 3], [4, 4, 4, 4] ], np.array([ [0.2, 0, 0.3, 0.1], [0.1, 0, 0.2, 0], [0.05, 0, 0, 0], [0, 0, 0, 0.05] ]))
```

Формат вывода

```
np.array([[0.2, 0.2, 0.3, 0.4], [0.6, 0.6, 1., 1.2], [0.45, 0.45, 0.6, 0.6], [0.2, 0.2, 0., 0.2]])
```

Примечания

Необходимо сдать файл `calc_expectations.py` с функцией `calc_expectations`. Циклы использовать запрещено!

С. Замена на среднее. 2021

Ограничение времени	0.4 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Написать функцию `replace_nan_to_means(X)`, принимающую матрицу `X`. Функция должна вернуть копию матрицы `X`, в которой все значения `nan` в каждом столбце заменены на среднее арифметическое остальных элементов столбца. В случае столбца из одних `nan` необходимо заменить все элементы столбца на нули. Исходная матрица `X` должна остаться неизменной!

Формат ввода

```
replace_nan_to_means(np.array([ [0, 1, 2], [np.nan, 1, np.nan], [5, 6, 7] ]))
```

Формат вывода

```
np.array([ [0, 1, 2], [2.5, 1, 4.5], [5, 6, 7] ])
```

Примечания

Необходимо сдать файл `replace_nan_to_means.py` с функцией `replace_nan_to_means`. Циклы использовать запрещено!

D. Поиск максимума. 2021

Ограничение времени	0.4 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Написать функцию `get_max_before_zero(x)`, возвращающую максимальный элемент в векторе `x` среди элементов, перед которыми стоит нулевой. Если подходящих элементов нет, функция должна возвращать `None`.

Формат ввода

```
get_max_before_zero(np.array([1, 2, 9, 8, 0, 5]))
```

Формат вывода

5

Примечания

Необходимо сдать файл `get_max_before_zero.py` с функцией `get_max_before_zero`. Циклы использовать запрещено!

Е. Кодирование длин серий. 2021

Ограничение времени	0.4 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Написать функцию `encode_rle(x)`, реализующую кодирование длин серий (Run-length encoding). По входному вектору `x` необходимо вернуть кортеж из двух векторов одинаковой длины. Первый содержит числа, а второй — сколько раз их нужно повторить.

Формат ввода

```
encode_rle(np.array([0, 0, 1, 1, 1, 2, 1]))
```

Формат вывода

```
(np.array([0, 1, 2, 1]), np.array([2, 3, 1, 1]))
```

Примечания

Необходимо отправить файл `encode_rle.py` с реализованной функцией `encode_rle`. Циклы использовать запрещено.

Г. Векторизация. 2021

К одной из задач **В-Е** напишите **три реализации**:

1. Полностью не векторизованную (без использования `numpy`)
2. Частично векторизованную
3. Полностью векторизованную

Сравните время работы трёх реализаций на данных разного размера, **постройте графики** зависимости времени работы алгоритмов от размера входных данных. Оформите и сдайте короткий `ipython notebook` с решением.