

Лабораторная работа №26 (2 часа)

Тема работы: «Разработка, отладка и испытание программ сортировки массивов»

1 Цель работы

Закрепить навык создания и работы с массивами больших данных на базе модуля NumPy.

2 Задание

Заполните массив случайными числами, перемешайте массив, возвести массив в степень, отсортировать массив.

3 Оснащение работы

Задание по варианту, ЭВМ, среда разработки **Python 3.7, IDLE**.

4 Основные теоретические сведения

Сортировка

`sort(a[, axis, kind, order])`

Возвращает отсортированную копию массива.

`lexsort(keys[, axis])`

Выполняет обратную устойчивую сортировку на основе указанных ключей (лексикографическая сортировка).

`argsort(a[, axis, kind, order])`

Возвращает индексы, сортирующие элементы исходного массива.

`ndarray.sort([axis, kind, order])`

Функция `sort()` реализованная в виде метода базового класса `ndarray`.

`msort(a)`

Возвращает отсортированную по первой оси копию массива.

`sort_complex(a)`

Возвращает отсортированную по последней оси копию массива комплексных чисел.

`partition(a, kth[, axis, kind, order])`

Возвращает копию исходного массива элементы которого разделены по указанному значению.

`argpartition(a, kth[, axis, kind, order])`

Возвращает массив индексов элементов исходного массива в их разбиении по указанному значению.

Поиск

`argmax(a[, axis, out])`

Возвращает индекс максимального значения вдоль указанной оси.

`nanargmax(a[, axis])`

Возвращает индекс максимального значения вдоль указанной оси с игнорированием значений *nan*

`argmin(a[, axis, out])`

Возвращает индекс минимального значения вдоль указанной оси.

`nanargmin(a[, axis])`

Возвращает индекс минимального значения вдоль указанной оси с игнорированием значений *nan*.

`argwhere(a)`

Возвращает индексы ненулевых элементов указанного массива.

`nonzero(a)`

Возвращает индексы ненулевых элементов массива.

`flatnonzero(a)`

Возвращает индексы ненулевых элементов в сжатом до одной оси представлении указанного массива.

`where(condition, [x, y])`

Возвращает элементы, которые могут выбираться из двух массивов в зависимости от условия.

`searchsorted(a, v[, side, sorter])`

Возвращает индексы в которые должны быть вставлены указанные элементы, что бы порядок сортировки был сохранен.

`extract(condition, arr)`

Возвращает элементы массива, которые удовлетворяют указанному условию.

Подсчет

`count_nonzero(a[, axis])`

Возвращает количество ненулевых элементов массива вдоль указанной оси.

Некоторые характеристики матриц

`linalg.eig(a)` - собственные значения и собственные векторы.

`linalg.norm(x[, ord, axis])` - норма вектора или оператора.

`linalg.cond(x[, p])` - число обусловленности.

`linalg.det(a)` - определитель.

`linalg.slogdet(a)` - знак и логарифм определителя (для избежания переполнения, если сам определитель очень маленький).

Системы уравнений

`linalg.solve(a, b)` - решает систему линейных уравнений $Ax = b$.

`linalg.tensorsolve(a, b[, axes])` - решает тензорную систему линейных уравнений $Ax = b$.

`linalg.lstsq(a, b[, rcond])` - метод наименьших квадратов.

`linalg.inv(a)` - обратная матрица.

Замечания:

`linalg.LinAlgError` – исключение, вызываемое данными функциями в случае неудачи (например, при попытке взять обратную матрицу от вырожденной).

Подробная документация, как всегда, на английском:
<https://docs.scipy.org/doc/numpy/reference/routines.linalg.html>

Массивы большей размерности в большинстве функций `linalg` интерпретируются как набор из нескольких массивов нужной размерности. Таким образом, можно одним вызовом функции проделывать операции над несколькими объектами.

```
>>>
>>> a = np.arange(18).reshape((2,3,3))
>>> a
array([[[ 0, 1, 2],
        [ 3, 4, 5],
        [ 6, 7, 8]],

       [[ 9, 10, 11],
        [12, 13, 14],
        [15, 16, 17]]])
>>> np.linalg.det(a)
array([ 0., 0.]
```

5 Порядок выполнения работы

1. Выделить ключевые моменты задачи.
2. Построить алгоритм решения задачи.
3. Запрограммировать полученный алгоритм.
4. Провести тестирование полученной программы.

6 Форма отчета о работе

Лабораторная работа № ____

Номер учебной группы _____

Фамилия, инициалы учащегося: _____

Дата выполнения работы: _____

Тема работы: _____

Цель работы: _____

Оснащение работы: _____

Результат выполнения работы: _____

7 Контрольные вопросы и задания

1. Как произвести сортировку массива средствами модуля NumPy?
2. Как выполнить поиск по массиву?
3. Как выполнить подсчет в массиве?

4. Перечислите исключения которые могут возникнуть при сортировке и поиске.

8 Рекомендуемая литература

Плас, Дж. В. Python для сложных задач. Наука о данных и машинное обучение / Дж.В. Плас. – СПб: Питер, 2018.

Прохоренок, Н.А. Python 3. Самое необходимое / Н.А Прохоренок, В.А. Дронов – СПб.: БВХ-Петербург, 2016.