



HCMUTE

TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

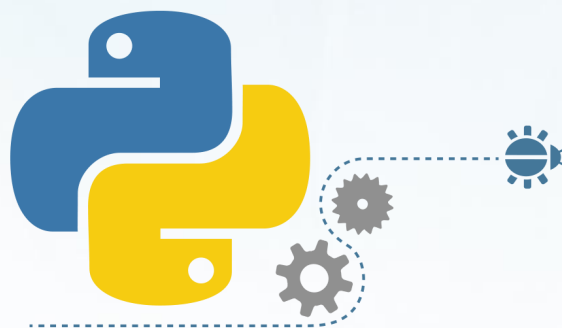
HCMC University of Technology and Education



KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN HỆ THỐNG THÔNG TIN

NHẬP MÔN LẬP TRÌNH PYTHON (IPPA233277)

BÀI TẬP PHÂN TÍCH DỮ LIỆU - THƯ VIỆN NUMPY



GV. Trần Quang Khải

1. Giới thiệu về phân tích dữ liệu
2. Tiến trình phân tích dữ liệu
3. Phân loại phân tích dữ liệu
4. Các gói phân tích dữ liệu
5. Thư viện numpy



1. Tạo một mảng numpy có kích thước 2×4 với toàn bộ giá trị bằng 0
2. Cho numpy array một chiều với giá trị ngẫu nhiên. Lọc và trả về mảng numpy gồm các giá trị số nguyên dương
3. Dùng hàm `argmax` trong numpy, tìm chỉ mục của số lớn nhất trong mảng một chiều
4. Tạo numpy array chứa các số lẻ từ 1 đến N, với N là số nguyên dương nhập bởi người dùng.
Gợi ý: dùng hàm `numpy.arange()` (tương tự hàm `range()` của Python. Sinh viên có thể dùng cú pháp ? hoặc google để biết thêm thông tin về hàm).
5. Tạo numpy array chứa 20 số thực cách đều nhau thuộc khoảng (a, b) với a, b nhập bởi người dùng.
Gợi ý: dùng hàm `numpy.linspace()` (có thể google cách dùng).
6. Tạo numpy array chứa 10 số thực ngẫu nhiên trong khoảng (a, b) với a, b nhập bởi người dùng.
Gợi ý: dùng `random.rand()` (tạo số thực ngẫu nhiên trong $[0,1)$) và cộng/trừ/nhân/chia với a, b để ra khoảng mong muốn.



7. Cho số lượng hàng hóa bán ra trong tuần của một cửa hàng được lưu trong một mảng 2D có shape (2, 7), với mỗi dòng tương ứng với buổi (sáng, chiều), và mỗi cột tương ứng với ngày trong tuần. Sinh viên tự cho giá trị các phần tử (có thể dùng `random.randint()`).

Hãy viết chương trình cho biết:

- Ngày bán được nhiều nhất tuần (theo tổng số hàng của cả 2 buổi). Gợi ý: dùng for trên cột.
- Thời điểm bán được nhiều nhất (buổi nào, ngày nào). Gợi ý: dùng for lồng nhau.
- Buổi nào có xu hướng bán được nhiều hàng hơn. Ví dụ: buổi sáng bán nhiều hơn buổi chiều trong 4 ngày thì kết luận buổi sáng. Còn nếu buổi sáng bán nhiều buổi chiều trong 3 ngày, 1 ngày bán bằng nhau, thì kết luận cả 2 buổi như nhau. Gợi ý: dùng for lồng nhau.



8. An được giao nhiệm vụ tổ chức thi tiếng Anh trên máy tính cho hai phòng với 07 bạn mỗi bên. Tuy nhiên, vì hôm ấy chỉ còn một phòng máy trống, An phải lựa chọn thí sinh để đảm bảo có nhiều nhất 07 bạn tham gia. An cũng biết được phòng 2 có thể thay bằng hình thức thi giấy. An đặt hai danh sách hai bên, chọn lựa giữa hai bạn vị trí đầu tiên, thứ hai, thứ ba,... đến hết. Tiêu chí chọn như sau:

- ✓ Giữa phòng 1 và 2, ưu tiên phòng 1.
- ✓ Nếu thí sinh phòng 1 xin thôi thi, chọn thí sinh còn lại.
- ✓ Nếu cả hai thí sinh đều thôi thi, gán “None” để bổ sung sau. Viết hàm nhận vào hai mảng numpy một chiều chứa số thứ tự thí sinh. Trả về mảng chứa danh sách thí sinh cuối cùng. Lưu ý rằng số thứ tự giữa hai phòng không trùng nhau và thí sinh thôi thi có số thứ tự âm. Ví dụ:

`room_1 = [1, 2, -3, 4, 5, 6, -7]`

`room_2 = [8, 9, 10, 11, 12, -13, -14]` Đáp án là: `[1, 2, 10, 4, 5, 6, None]`

Tên hàm: `combine_rooms(room_1, room_2)`



9. Cho hai ma trận $A, B \in R^{m \times n}$. Phép cộng ma trận được thực hiện như sau:

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & \cdots & a_{1n} + b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & \cdots & a_{mn} + b_{mn} \end{bmatrix}$$

Khi thực hiện trên máy tính, nếu $B \in R^m$ là một vector, phần mềm sẽ thực hiện broadcast vector bằng cách tạo ma trận mới với mỗi cột chính là B . Ví dụ:

$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} + \begin{bmatrix} 6 \\ 7 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} + \begin{bmatrix} 6 & 6 \\ 7 & 7 \end{bmatrix} = \begin{bmatrix} 8 & 9 \\ 11 & 12 \end{bmatrix}$$

Mô phỏng lại broadcast bằng một hàm nhận một vector numpy array và số cột. Hàm trả về kết quả broadcast. Chẳng

hạn, với $\begin{bmatrix} 6 \\ 7 \end{bmatrix}$ và số cột là 3, kết quả là $\begin{bmatrix} 6 & 6 & 6 \\ 7 & 7 & 7 \end{bmatrix}$

Tên hàm: **broadcast(vec, n)**



10. Ma trận chuyển vị (transpose) được tạo ra bằng cách lật ma trận ban đầu theo trục chéo

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \quad \mathbf{A}^T = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

Cho ma trận mảng numpy, trả về chuyển vị của ma trận đó

Tên hàm: `transpose(mat)`

11. Tích ma trận và tích hadamard là hai phép tính phổ biến. Hãy viết chương trình nhận hai ma trận đầu vào, in trên màn hình tích của chúng. Nếu không thể tính, hãy in “Khong co tich ma tran” hoặc “Khong co tich Hadamard”

Tên hàm: `product(mat_a, mat_b)`

12. Viết hàm nhận tham chiếu là ma trận và số chỉ mục của cột. Thay toàn bộ giá trị cột đó là 1 rồi trả về kết quả.

Tên hàm: `replace_col(mat, col_ind)`



- ✓ Họ tên : **Trần Quang Khải**
- ✓ Email : **khaitq@hcmute.edu.vn**
- ✓ Zalo (mã Qr)

