AI VIET NAM – Note from All-in-One Course

Học Numpy Qua Các Ví Dụ

(Đinh Quang Vinh)

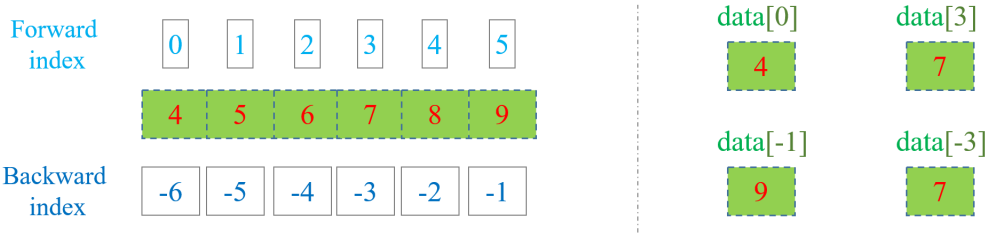
Ngày 8 tháng 8 năm 2023

Câu 1: Import gói Numpy và in ra phiên bản của nó

```python
# Python code (aivietnam)
import numpy as np
print(np.version.version)
```

```
================ Output ===============
1.25.0
=======================================
```

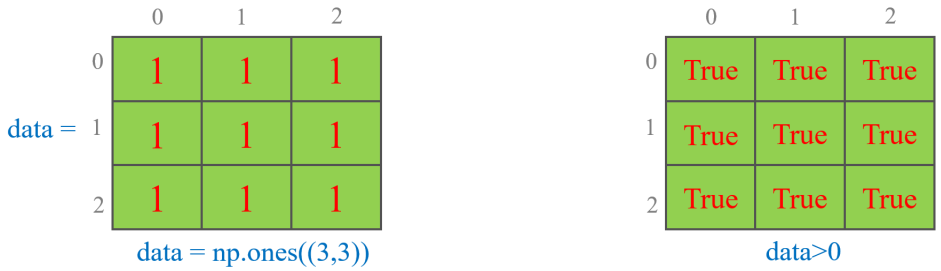
Câu 2: Tạo mảng một chiều từ 4 đến 9




```python
# Python code (aivietnam)
import numpy as np

data = np.arange(4, 10)
print(data)
```

```
================ Output ===============
[4 5 6 7 8 9]


=======================================
```

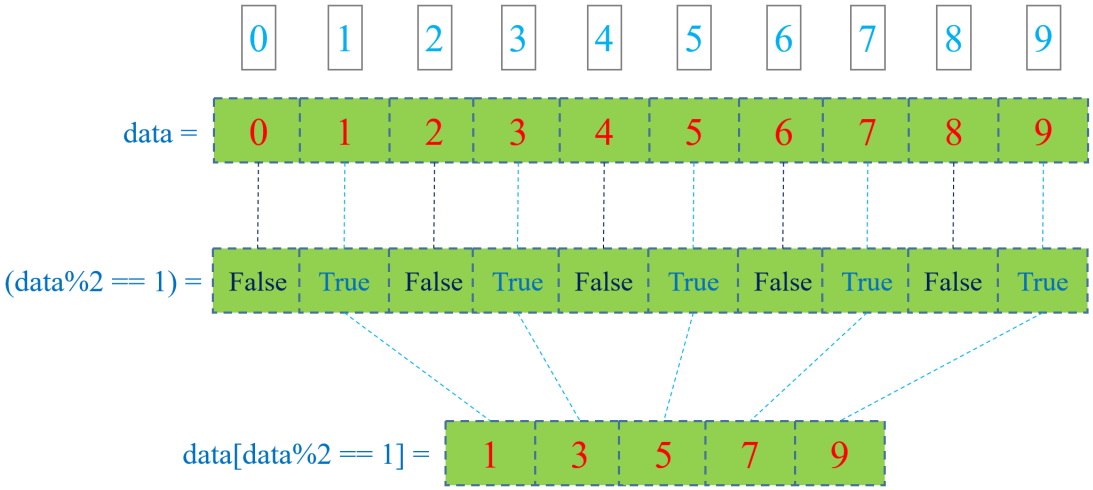
Câu 3: Tạo một mảng boolean 3x3 với tất cả giá trị là True



```python
# Python code (aivietnam)
import numpy as np

# way 1
data1 = np.ones((3,3)) > 0
print(f'{data1} \n----------------------')

# way 2
data2 = np.ones((3,3), dtype=bool)
print(f'{data2} \n----------------------')

# way 3
data3 = np.full((3,3), True, dtype=bool)
print(f'{data3} \n----------------------')
```

```
================ Output ===============
[[ True  True  True]
 [ True  True  True]
 [ True  True  True]]
--------------------
[[ True  True  True]
 [ True  True  True]
 [ True  True  True]]
--------------------
[[ True  True  True]
 [ True  True  True]
 [ True  True  True]]
--------------------
=======================================
```

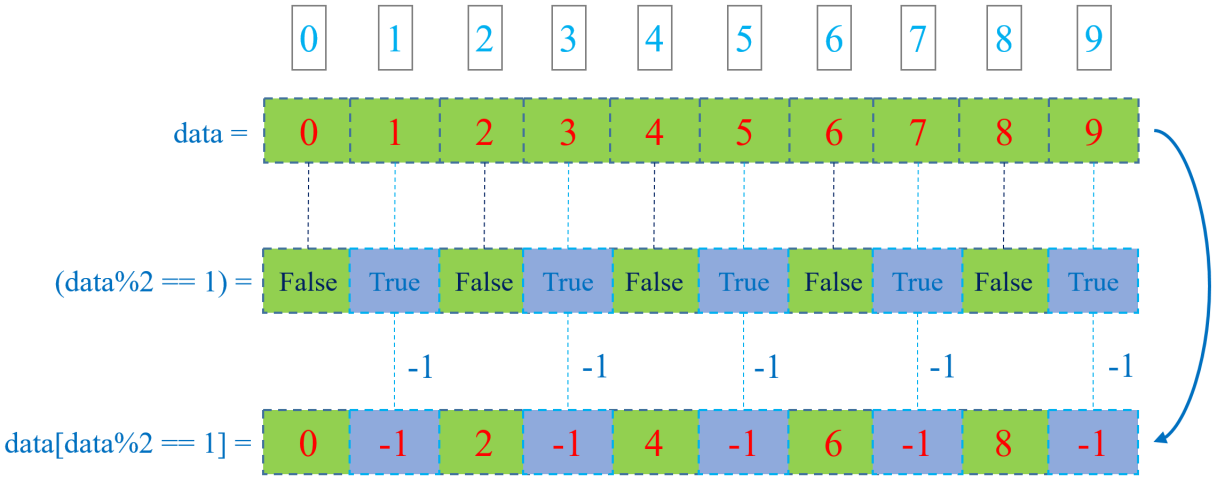
Câu 4: Lấy những phần tử mà thoả mãn một điều kiện cho trước của mảng một chiều




```python
# Python code (aivietnam)
import numpy as np

# create an ndarray from 0 to 9
data = np.arange(0, 10)
print(data)

# Find odd numbers
data_odd = data[data%2 == 1]
print(data_odd)
```

```
================ Output ===============
[0 1 2 3 4 5 6 7 8 9]
[1 3 5 7 9]
=======================================
```

Câu 5: Thay thế phần tử thoả mãn điều kiện cho trước bằng một giá khác




```python
# Python code (aivietnam)
import numpy as np

# create an ndarray from 0 to 9
data = np.arange(0, 10)
print(data)

# replace odd numbers by -1
data[data%2 == 1] = -1
print(data)
```

```
================ Output ================
[0 1 2 3 4 5 6 7 8 9]
[ 0 -1  2 -1  4 -1  6 -1  8 -1]
========================================
```

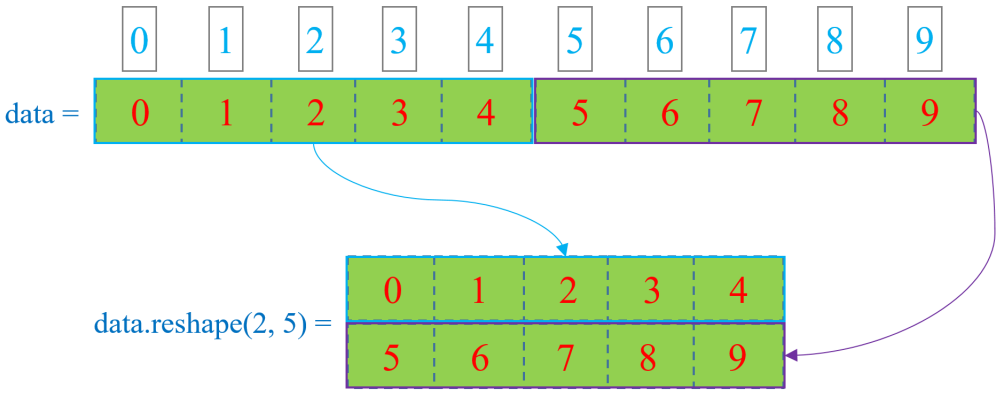
```python
# Python code (aivietnam)
import numpy as np

# create an ndarray from 0 to 9
data = np.arange(0, 10)
print(data)

# replace odd numbers by -1
out = np.where(data%2 == 1, -1, arr)
print(out)
```

```
================ Output ================
[0 1 2 3 4 5 6 7 8 9]
[0 -1  2 -1  4 -1  6 -1  8 -1]
========================================
```

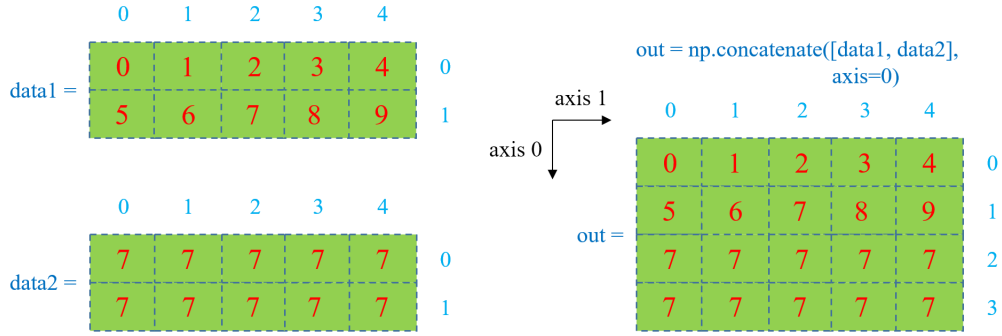
Câu 6: Chuyển định dạng (shape) của một ndarray. Chuyển mảng một chiều thành mảng hai chiều

```python
# Python code (aivietnam)
import numpy as np

# create a 1D ndarray from 0 to 9
data = np.arange(10)
print(data)

# reshape data to 2 rows and 5 columns
data_2d = data.reshape(2, 5)
print(data_2d)
```

```
================ Output ================
[0 1 2 3 4 5 6 7 8 9]
[[0 1 2 3 4]
 [5 6 7 8 9]]
========================================
```



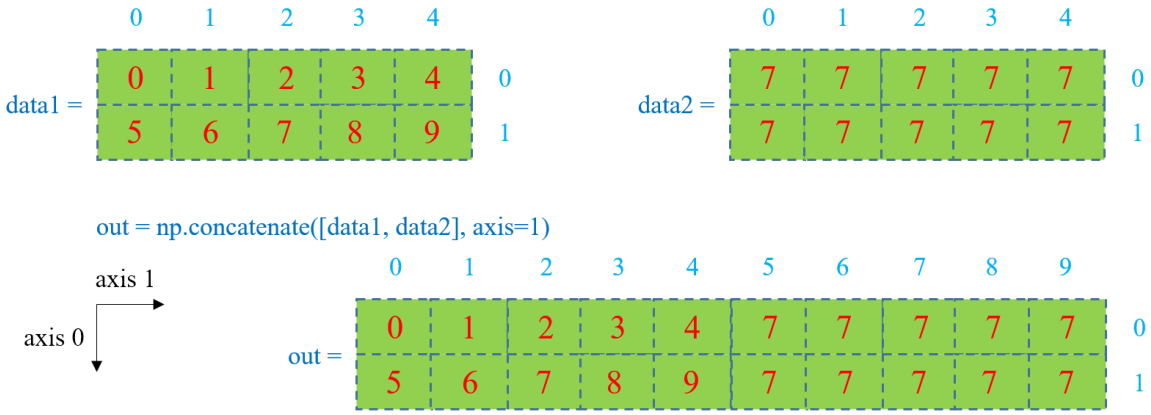
Câu 7: Xếp chồng 2 mảng theo chiều dọc




```python
# Python code (aivietnam)
import numpy as np

data1 = np.arange(10).reshape(2, -1)
print(data1)

data2 = np.repeat(7, 10).reshape(2, -1)
print(data2)

# Way 1 :
out1 = np.concatenate([data1, data2],
                      axis=0)
print(out1)

# Way 2 :
out2 = np.vstack([data1, data2])
print(out2)

# Way 3 :
out3 = np.r_[data1, data2]
print(out3)
```

```
================= Output ================
[[0 1 2 3 4]
 [5 6 7 8 9]]
[[7 7 7 7 7]
 [7 7 7 7 7]]
[[0 1 2 3 4]
 [5 6 7 8 9]
 [7 7 7 7 7]
 [7 7 7 7 7]]
[[0 1 2 3 4]
 [5 6 7 8 9]
 [7 7 7 7 7]
 [7 7 7 7 7]]
[[0 1 2 3 4]
 [5 6 7 8 9]
 [7 7 7 7 7]
 [7 7 7 7 7]]


=========================================
```

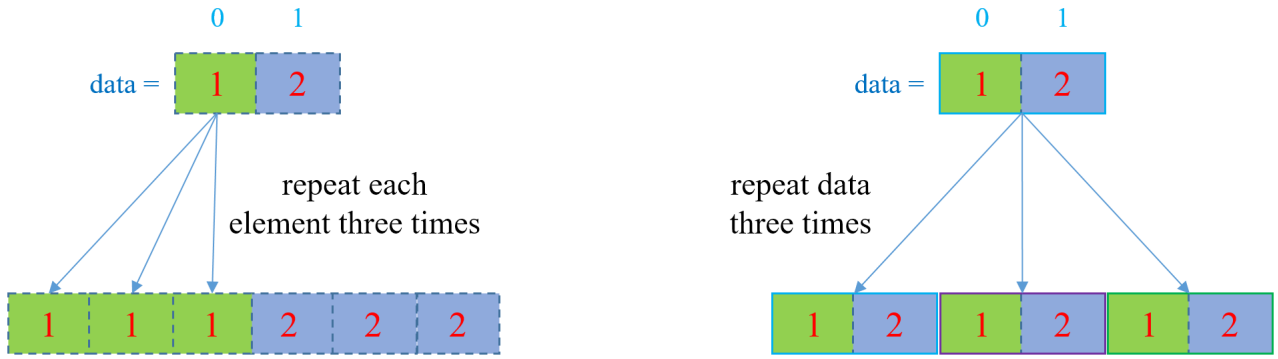
Câu 8: Xếp chồng 2 mảng theo chiều ngang




```python
# Python code (aivietnam)
import numpy as np

data1 = np.arange(10).reshape(2, -1)
print(data1)

data2 = np.repeat(7, 10).reshape(2, -1)
print(data2)

# Way 1
out1 = np.concatenate([data1, data2],
                      axis=1)
print(out1)

# Way 2
out2 = np.hstack([data1, data2])
print(out2)

# Way 3
out3 = np.c_[data1, data2]
print(out3)
```

```
================= Output ================
[[0 1 2 3 4]
 [5 6 7 8 9]]
[[7 7 7 7 7]
 [7 7 7 7 7]]
[[0 1 2 3 4 7 7 7 7 7]
 [5 6 7 8 9 7 7 7 7 7]]
[[0 1 2 3 4 7 7 7 7 7]
 [5 6 7 8 9 7 7 7 7 7]]
[[0 1 2 3 4 7 7 7 7 7]
 [5 6 7 8 9 7 7 7 7 7]]
=========================================
```

Câu 9: Lặp data với repeat() và tile()



```python
# Python code (aivietnam)
import numpy as np

data = np.array([1, 2])
print(data)

# repeat each element three times
out1 = np.repeat(data, 3)
print(out1)

# repeat data three times
out2 = np.tile(data, 3)
print(out2)
```

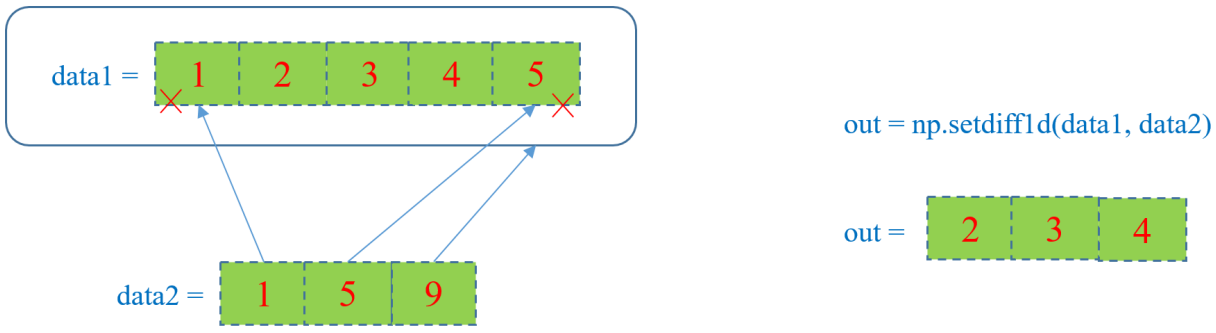
```
================ Output ===============
[1 2]
[1 1 1 2 2 2]
[1 2 1 2 1 2]


=======================================
```

Câu 10: Lấy phần tử chung của 2 mảng


```python
# Python code (aivietnam)
import numpy as np

data1 = np.array([1, 2, 3, 2])
data2 = np.array([7, 2, 1, 8])
print(data1)
print(data2)

out = np.intersect1d(data1, data2)
print(out)
```

```
================ Output ===============
[1 2 3 2]
[7 2 1 8]
[1 2]


=======================================
```

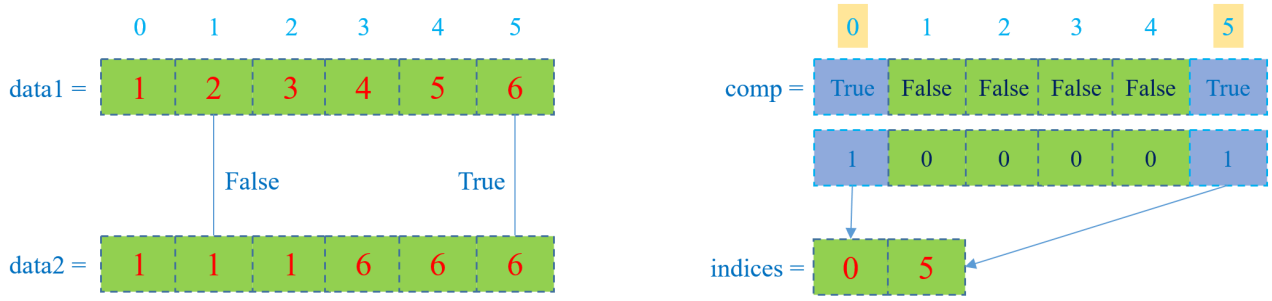
Câu 11: Xoá phần tử từ một mảng mà tồn tại trong một mảng khác



```python
# Python code (aivietnam)
import numpy as np

data1 = np.array([1, 2, 3, 4, 5])
data2 = np.array([1, 5, 9])

out = np.setdiff1d(data1, data2)
print(out)
```

```
================ Output ===============
[2 3 4]

=======================================
```

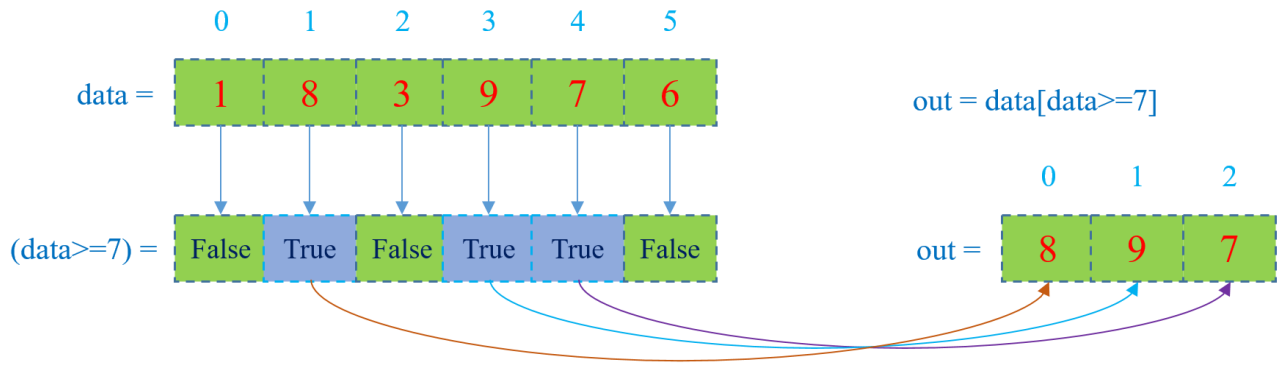
Câu 12: Lấy tất cả vị trí nơi giá trị các phần tử của hai mảng giống nhau




```python
# Python code (aivietnam)
import numpy as np

# create data1 v   data2
data1 = np.array([1, 2, 3, 4, 5, 6])
data2 = np.array([1, 1, 1, 6, 6, 6])

# compare the two array
comp = data1==data2

# get indices whose elements are not zero
indices = comp.nonzero()
print(indices)
```

```
================ Output ===============
(array([0, 5], dtype=int64),)

=======================================
```

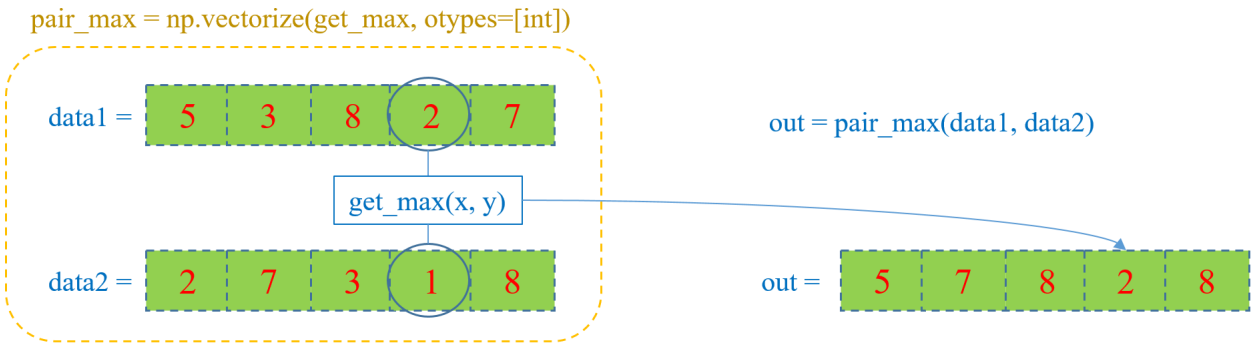
Câu 13: Lấy tất cả các giá trị trong một phạm vi cho trước



```python
# Python code (aivietnam)
import numpy as np

data = np.array([1, 8, 3, 9, 7, 6])
print(data)

# Way 1
indices = np.where(data>=7)
out1 = data[indices]
print(out1)

# Way 2
out2 = data[data>=7]
print(out2)
```

```
================ Output ===============
[1 8 3 9 7 6]
[8 9 7]
[8 9 7]
=======================================
```

Câu 14: Áp dụng một hàm user-defined cho ndarray dùng np.vectorize(). Áp dụng hàm get_max() cho hai mảng ndarray



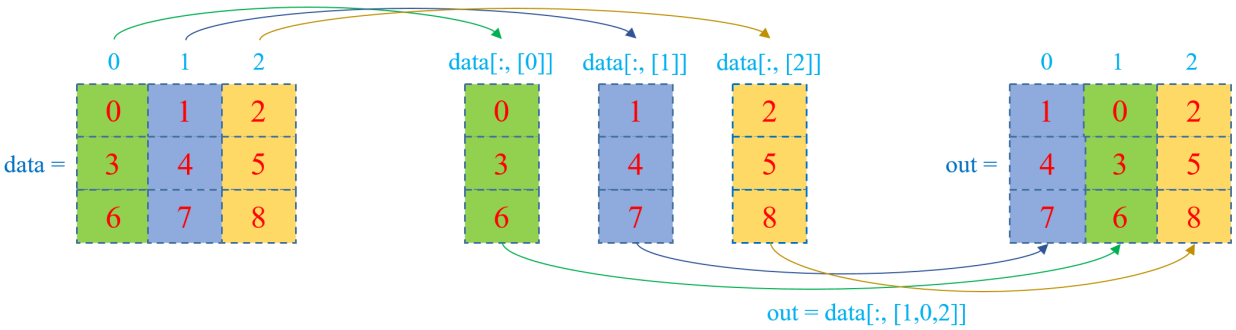

```python
# Python code (aivietnam)
import numpy as np

# get larger value
def get_max(x, y):
    if x >= y :
        return x
    else:
        return y

# vectorize the function
pair_max = np.vectorize(get_max,
                        otypes=[int])

# create data1 and data2
data1 = np.array([5, 3, 8, 2, 7])
data2 = np.array([2, 7, 3, 1, 8])

# use pair_max as a function
out1 = pair_max(data1, data2)
print(out1)
```

```
================ Output ===============
[5 7 8 2 8]
=======================================
```

```python
# Python code (aivietnam)
import numpy as np

data1 = np.array([5, 3, 8, 2, 7])
data2 = np.array([2, 7, 3, 1, 8])

# Way 2: Using the maximum() function
out2 = np.maximum(data1, data2)
print(out2)

# Way 3: Using the where() function
out3 = np.where(data1>data2, data1, data2)
print(out3)
```

```
================ Output ===============
[5 7 8 2 8]
[5 7 8 2 8]
=======================================
```

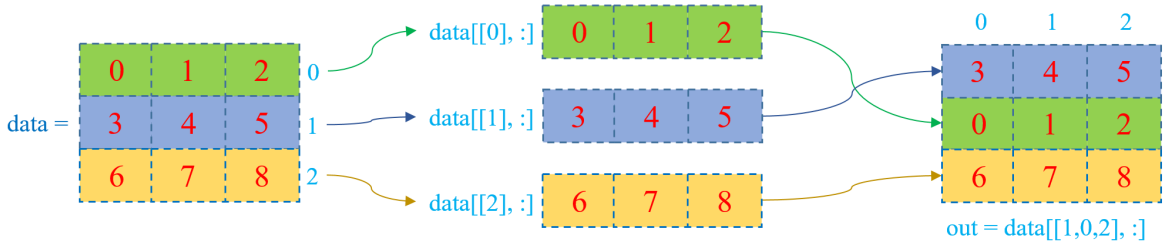
Câu 15: Hoán đổi các cột trong mảng hai chiều




```python
# Python code (aivietnam)
import numpy as np

# create a 3x3 matrix
data = np.arange(9).reshape(3,3)
print(data, '\n')

# A new matrix is constructed by the
# columns [1,0,2] from data
out = data[:, [1,0,2]]
print(out)
```

```
================ Output ===============
[[0 1 2]
 [3 4 5]
 [6 7 8]]

[[1 0 2]
 [4 3 5]
 [7 6 8]]

=======================================
```

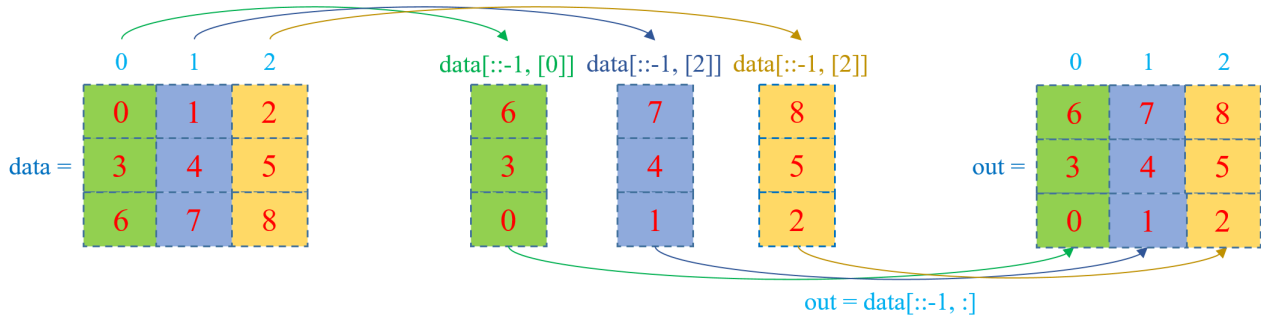
Câu 16: Hoán đổi các dòng trong mảng hai chiều



```python
# Python code (aivietnam)
import numpy as np

# create a 3x3 matrix
data = np.arange(9).reshape(3,3)
print(data, '\n')

# A new matrix is constructed by the
# rows [1,0,2] from data
out = data[[1,0,2], :]
print(out)
```

```
================ Output ===============
[[0 1 2]
 [3 4 5]
 [6 7 8]]

[[3 4 5]
 [0 1 2]
 [6 7 8]]


=======================================
```

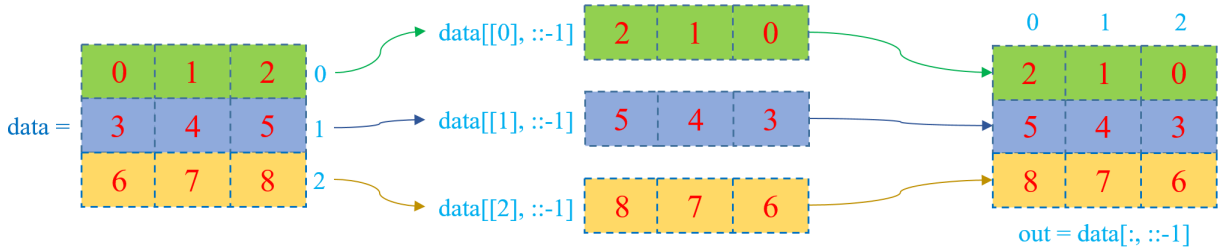
Câu 17: Đảo ngược các phần tử của các cột trong mảng hai chiều




```python
# Python code (aivietnam)
import numpy as np

# create a 3x3 matrix
data = np.arange(9).reshape(3,3)
print(data, '\n')

# reverse each column
out = data[::-1, :]
print(out)
```

```
================ Output ===============
[[0 1 2]
 [3 4 5]
 [6 7 8]]

[[6 7 8]
 [3 4 5]
 [0 1 2]]

=======================================
```

Câu 18: Đảo ngược các phần tử của các hàng trong mảng hai chiều



```python
# Python code (aivietnam)
import numpy as np

# create a 3x3 matrix
data = np.arange(9).reshape(3,3)
print(data, '\n')

# reverse each row
out = data[:, ::-1]
print(out)
```

```
================ Output ===============
[[0 1 2]
 [3 4 5]
 [6 7 8]]

[[2 1 0]
 [5 4 3]
 [8 7 6]]

=======================================
```

Câu 19: Tạo mảng hai chiều chứa số ngẫu nhiên (kiểu số lẻ)

data1 =

9.36	9.05	8.22
5.11	8.58	6.60

np.random.uniform(5, 10, size=(2, 3))

data2 =

0.71	0.48	0.96
0.27	0.54	0.37

np.random.random([2, 3])

```python
# Python code (aivietnam)
import numpy as np

# create a 2x3 matrix (from 5 to 10)
data1 = np.random.uniform(5, 10,
                          size=(2, 3))
print(data1, '\n')

# create a 2x3 matrix (from 0 to 1)
data2 = np.random.random([2, 3])
print(data2)
```

```
================ Output ===============
[[9.36267749 9.05352382 8.22208754]
 [5.1182297  8.58622514 6.60564768]]

[[0.71150894 0.48885357 0.96791849]
 [0.2763089  0.54731615 0.37406768]]

=======================================
```