Lab-07

March 22, 2023

1 Questions

1. Write a NumPy program to create a structured array from given student name, height, age and their data types. Now sort by age, then height if age are equal.
2. Write a NumPy program to get the indices of the sorted elements of a given array.
3. Write a NumPy program to test whether none of the elements of a given array is zero.
4. Write a NumPy program to swap rows and columns of a given array in reverse order.
5. Write a NumPy program to multiply two given arrays of same size element-by-element.
6. Write a NumPy program to convert a given list into an array, then again convert it into a list. Check initial list and final list are equal or not.
7. Write a NumPy program to convert a list of numeric value into a one-dimensional NumPy array.
8. Write a NumPy program to convert a list and tuple into arrays.
9. Write a NumPy program to get the unique elements of an array.
10. Write a NumPy program to construct an array by repeating.

2 Answers

Dipankar Das 20051554

2.1 Write a NumPy program to create a structured array from given student name, height, age and their data types. Now sort by age, then height if age are equal.

```python
[51]: import numpy as np

class Student:
  def __init__(self, name: str, height:float, age:int):
    self.name = name
    self.height = height
    self.age = age
  def display(self):
    print(f"Name: {self.name},Height: {self.height},Age: {self.age}")

a = np.array([Student("Dipankar", 7.132, 21),
              Student("Demo", 3242.32, 23),
              Student("edwdwe", 2342.324, 23)], dtype=Student)
```

```
for stu in a:
  stu.display()
```

```
Name: Dipankar,Height: 7.132,Age: 21
Name: Demo,Height: 3242.32,Age: 23
Name: edwdwe,Height: 2342.324,Age: 23
```

2.2 Write a NumPy program to get the indices of the sorted elements of a given array.

[28]:
```
A = np.array([2,3,4,6,1], dtype=np.int32)
print(A)

print(f"sorted element index diff from org => {np.argsort(A)}")
```

```
[2 3 4 6 1]
sorted element index diff from org => [4 0 1 2 3]
```

2.3 Write a NumPy program to test whether none of the elements of a given array is zero.

[31]:
```
arr = np.array([32,2,10,123,131,130,0], dtype=np.int32)
flag = False
for item in arr:
  if item == 0:
    flag = True
    break
if not flag:
  print("Zero is absent")
else:
  print("Zero is present")
```

```
Zero is present
```

2.4 Write a NumPy program to swap rows and columns of a given array in reverse order.

[35]:
```
m = int(input("Enter no of rows"))
n = int(input("Enter no of columns"))

arr = np.zeros((m,n), dtype=np.int32)

for i in range(m):
  for j in range(n):
    arr[i,j] = np.int32(input("Enter the element"))

arr = arr[::-1, :]
```

2

```
arr = arr[:, ::-1]

print(arr)
```

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

2.5 Write a NumPy program to multiply two given arrays of same size element-by-element.

```
[36]: mult1 = np.array([[1,2],
                        [3,4]], dtype=np.int32)
      mult2 = np.array([[1,2],
                        [3,4]], dtype=np.int32)

      print(mult1 * mult2)
```

```
[[ 1  4]
 [ 9 16]]
```

2.6 Write a NumPy program to convert a given list into an array, then again convert it into a list. Check initial list and final list are equal or not.

```
[39]: lst = list[34,4,1,0]
      print(lst)
      arrLst = np.array(lst)
      print(arrLst)
      print(arrLst.tolist() == lst)
```

```
list[34, 4, 1, 0]
list[34, 4, 1, 0]
True
```

2.7 Write a NumPy program to convert a list of numeric value into a one-dimensional NumPy array.

```
[41]: lst = list[34,4,1,0]
      print(lst)
      np.array(lst)
```

```
list[34, 4, 1, 0]
```

```
[41]: array(list[34, 4, 1, 0], dtype=object)
```

2.8   Write a NumPy program to convert a list and tuple into arrays.

```
[46]: lst = list[34,45,13,3]
      tup = (34,234,234)
      # print(tup)
      # print(lst)
      print(np.array(lst))
      print(np.array(tup))
```

```
list[34, 45, 13, 3]
[ 34 234 234]
```

2.9   Write a NumPy program to get the unique elements of an array.

```
[52]: arr = np.array([1,2,2,3,4,4], dtype=np.int32)
      print(np.unique(arr))
```

```
[1 2 3 4]
```

2.10   Write a NumPy program to construct an array by repeating.

```
[55]: a = [1, 2, 3, 4]
      print(a)
      x = np.tile(a, 2)
      print(x)
```

```
[1, 2, 3, 4]
[1 2 3 4 1 2 3 4]
```