暨南大学本科实验报告专用纸

课程名称 python 实验 成绩评定

实验项目名称 序列型数据类型 指导教师 林聪

实验项目编号 实验项目类型 实验地点

学生姓名 王志涛 学号 2021102259

学院 智能科学与工程 系 专业 人工智能

实验时间 2022 年 4 月 20 日 午～ 5 月 10 日 午

一、 实验目的

通过自拟案例，初始化并测试各类的序列型数据类型.

二、 实验原理

Python 可以实现测试

三、 实验过程

1．列表嵌套：

```
l=[[0,1],[2,3],[[4,5],[6,7]]]
print(l[2][0][0])
```

集合

```
my_name={"wang","zhi","tao"}
for x in my_name:
    print(x)
```

字典：

```
my_happy={
    "wang":123,
    "zhi":456,
    "tao":678
}
for x,y in my_happy.items():
    print(x,y)
```

OrderedDict

```
import collections
print("Regular dictionary")
d={}
d['b']='B'
d['a']='A'
d['c']='C'
for k,v in d.items():
    print (k,v)
print ("\nOrder dictionary")
d1 = collections.OrderedDict()
d1['a'] = 'A'
d1['b'] = 'B'
d1['c'] = 'C'
d1['1'] = '1'
d1['2'] = '2'
for k,v in d1.items():
    print (k,v)
```

函数 len()：

```
my_name="wang zhi tao"
print(len(my_name))
```

3. 列表的相关方法：

.append()：

```
my_name=["wang","zhi","tao"]
my_name.append("love")
my_name.append("you")
print(my_name)
```

Del：

```
my_name=["wang","zhi","tao"]
del my_name[0]
print(my_name)
```

pop()：

```
my_name=["wang","zhi","tao"]
my_name_pop=my_name.pop(1)
print(my_name)
print(my_name_pop)
```

字典的相关方法：

.keys()：

```
dishes = {'eggs': 2, 'sausage': 1, 'bacon': 1, 'spam': 500}
keys=dishes.keys()
print(list(keys))
```

.values()：

```
dishes = {'eggs': 2, 'sausage': 1, 'bacon': 1, 'spam': 500}
values=dishes.values()
print(list(values))
```

.items()：

```
dishes = {'eggs': 2, 'sausage': 1, 'bacon': 1, 'spam': 500}
for v,k in dishes.items():
    print(v,k)
```

5. 安装并了解 Numpy 库中的 ndarray 数据类型. 通过下面语句初始化一个 ndarray 对象：

```
import numpy
arr = numpy.array([[1,2,3],[4,5,6]])
print(arr)
```

输出：

```
[[1 2 3]
 [4 5 6]]
```

Arr.ndim:

ndim 返回的是数组的维度，返回的只有一个数，该数即表示数组的维度

arr.shape:

shape：表示各位维度大小的元组。返回的是一个元组

arr.dtype:

dtype：一个用于说明数组数据类型的对象。返回的是该数组的数据类型。

Arr.size:

Size:数组所含元素总数

```
import numpy
arr = numpy.array([[1,2,3],[4,5,6]])
a=arr.ndim
b=arr.shape
c=arr.size
d=arr.dtype
print(a)
print(b)
print(c)
print(d)
```

输出为：

```
2
(2, 3)
6
int32
```

6.行列式求值

```
import numpy
import numpy as np
arr = numpy.random.randint(low=0, high=10, size=(8,8), dtype='uint8')
h=numpy.linalg.det(arr)
print("正确答案:"+str(h))
def det(m):
    if len(m) <= 0:
        return None
    elif len(m) == 1:
        return m[0][0]
    else:
        s = 0
        for i in range(len(m)):
            row=m[1:]
            n = np.delete(row, i, axis=1)
            s += m[0][i] * det(n) * (-1) ** (i % 2)
        return s
print("你的答案:"+str(det(arr)))
```

思考题：

1. 什么是 ndarray 类型：

ndarray 对象是用于存放同类型元素的多维数组。

2. 如果一个 list 存储 1-100 的整数与一个 ndarray 向量中存储 1-100 有哪些区别?

python 中的 list 是 python 的内置数据类型，list 中的数据类不必相同的，而 array 的中的类型必须全部相同。

3. 比较 list 和 ndarray 各有哪些相对优势?

Ndarray 效率更高，速度更快。

Ndarray 必须是相同的数据类型，而 list 则不需要

Ndarray 数组之间可以进行矢量计算

4. 什么是惰性序列(lazy sequence)? 举出两个例子

Python 的 iterator 是一个惰性序列，意思是表达式和变量绑定后不会立即进行求值，而是当你用到其中某些元素的时候才去求某元素对的值。 惰性是指，你不主动去遍历它，就不会计算其中元素的值。

5. 惰性序列有哪些好处?

不需要耗用太多资源，可以实现无限序列的表示。

6. 给定一个迭代器，变量名 iter_A,，执行 list(iter_A)后，iter_A 是否能够再用于迭代? 为什么?

不可以，实践检验真理：

```
lists={10,20,30,40,50}
iter_A=iter(lists)
h=list(iter_A)
for i in iter_A:
    print(i,end="")
    print("\n")
```

四、实验结果

了解了各种列表和字典的函数，也实现了用 python 计算行列式的值。

五、 实验总结

这次实验的行列式求值花费时间较长，学会了许多关于 numpy 库的相关知识。