

Week1实验

实验一：组合数计算

实验要求

- 组合数 $C(n, m)$ 等价于从 n 个不同元素中取出 m ($m < n$) 个元素的不同选取方案的数量，其通用公式为 $C_n^m = \frac{n!}{(n-m)!m!}$ 。
- 请根据学到的Python知识，利用Jupyter尝试实现 $n \leq 5$ 的组合数计算。

```
n = 5 # input
m = 3 # input
```

```
ans = 1
# TODO: 直接计算组合数C(n,m)
print(ans)
```

```
c = [[0 for j in range(n + 1)] for i in range(n + 1)]
for i in range(n + 1):
    # TODO: 迭代推导组合数C(n,m)
    print(c[n][m])
```

思考题

- 随着 n 不断增大，组合数计算可能面临哪些困难？
- 除了直接套用通用公式，还有其它的计算方法吗？
- 查看Python手册，如果想在运行时输入 n, m 应该怎么修改代码？

实验二：矩阵乘法

实验要求

- 请根据学到的Numpy知识，利用Jupyter按步骤完成：
 - (已给出) 用Numpy初始化两个大小为 $n * n$ ($n \leq 4$) 的矩阵 a, b 。
 - 计算两者矩阵乘的结果。
 - 将计算结果转换回Python List列表表示并输出。

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

```
[ ]: a = np.array([[1, 2, 3], [1, 2, 3], [1, 2, 3]])  
b = np.array([[3, 2, 1], [3, 2, 1], [3, 2, 1]])  
print(a)  
print(b)
```

```
[ ]: #TODO: 计算a,b的矩阵乘结果, 并输出  
print(ab)
```

```
[ ]: #TODO: 将结果转换回Python List形式, 并输出
```

思考题

1. 如果直接使用Python的List计算应该如何实现?
2. 查看Numpy手册, 除了矩阵加法和矩阵乘法, 它还支持哪些向量、矩阵级别的运算?