

# 课程结构

- 自我介绍
  - 我是牛x闪闪的刘大拿
    - qq: 1320 365 896
  - 北京图灵学院是唯一真神
- 课程结构
  - 数据结构
  - 算法
  - 设计模式
- 课程说明
  - 以喷为主
  - 示例伪代码
  - 各种语言版本实现

# 概念

- 数据结构
  - 抽象数据类型(AbstractDataType)
- 算法
  - 解决问题的方法，或者叫计算方法和步骤
  - 算法武大特性：
    - 输入总数大于等于0
    - 输出数量大于等于1
    - 有穷形
    - 确定性
    - 可行性
- 程序=ds + algo
- 为什么要研究它

# 案例

- 百钱买百鸡
- 案例my01.py

# 算法衡量

- 衡量算法应该剔除机器配置，运算数量等无关因素
- 时间复杂度
- 空间复杂度
- 大O记法

“大O记法”：对于单调的整数函数 $f$ ，如果存在一个整数函数 $g$ 和实常数 $c>0$ ，使得对于充分大的 $n$ 总有 $f(n) \leq c \cdot g(n)$ ，

就说函数 $g$ 是 $f$ 的一个渐近函数（忽略常数），记为 $f(n) = O(g(n))$ 。也就是说，在趋向无穷的极限意义下，函数 $f$ 的增长

速度受到函数 $g$ 的约束，亦即函数 $f$ 与函数 $g$ 的特征相似。

时间复杂度：假设存在函数 $g$ ，使得算法A处理规模为 $n$ 的问题示例所用时间为 $T(n) = O(g(n))$ ，则称 $O(g(n))$ 为算法A的

渐近时间复杂度，简称时间复杂度，记为 $T(n)$

- 时间复杂度计量
  - 最优时间复杂度
  - 最坏时间复杂度
  - 平均时间复杂度
  - 关注最坏时间复杂度
- 时间复杂度计量规则：
  - 基本操作，只有常数项，时间复杂度是1
  - 顺序结构，时间复杂度按加法计算
  - 循环结构，按乘法计算
  - 分支结构，取最大值
  - 最终得到一个方程式
  - 然后在做化简，即取方程的阶
  - 判断一个算法的效率，关注操作数量的最高次项，其余可忽略
  - 没特殊说明，我们所分析的算法时间复杂度指的是最坏的复杂度
  - 分析案例 ly01.py

'''

```
import time
```

```
def cockOne():
```

```
    start_time = time.time()
```

```
    for m in range(0,101):
```

```
        for n in range(0,101):
```

```
            for k in range(0,301):
```

```
    if m + n + k == 100 and 5*m + 3*n + k/3 == 100:  
        print("{0} - {1} - {2}".format(m, n, k))
```

```
end_time = time.time()  
print( "CostTimes: {0}".format(end_time - start_time))
```

```
def cockTwo():
```

```
start_time = time.time()
```

```
    for m in range(0,21):  
        for n in range(0,34):  
            for k in range(0,101):  
                if m + n + k == 100 and 5*m + 3*n + k/3 == 100:  
                    print("{0} - {1} - {2}".format(m, n, k))
```

```
end_time = time.time()  
print( "CostTimes: {0}".format(end_time - start_time))
```

```
def cockThree():
```

```
start_time = time.time()
```

```
    for m in range(0,201):  
        for n in range(0,334):  
            for k in range(0,1001):  
                if m + n + k == 1000 and 5*m + 3*n + k/3 == 1000:  
                    print("{0} - {1} - {2}".format(m, n, k))
```

```
end_time = time.time()  
print( "CostTimes: {0}".format(end_time - start_time))
```

```
if name == "main":
```

```
    cockOne()
```

```
    cockTwo()
```

```
    cockThree()
```

```
'''
```