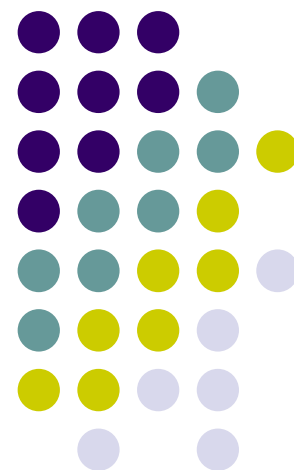


面向科学问题求解的编程实践





课程信息

- 面向科学问题求解的编程实践
 - 学时：30/20；周五（6，7）5201
 - 15次课
 - 上机时间：待定
 - 主讲：孙广中
gzsun@ustc.edu.cn
 - 助教：何钰、徐中天、王海林



2021 面向科学问题...

群号：689778055



扫一扫二维码，加入群聊。





课程信息

- 这是一门面向全校本科生的计算机编程实践课程。该课程面向**有一定编程知识的本科生**（已学过一门程序设计语言），通过有针对性的几个专题编程训练，进行问题求解的实践，加深学生对问题求解与程序设计的理解与认识，使得学生具有通过编写一定规模的程序求解问题的实践能力。作为一门以实践为主的课程，采用以**课堂内讲授与课堂外实践相结合**的授课形式。



课程信息

- 2020 年春季课程内容，40/40
 - 程序设计语言 (3/15)
 - Python（面向对象、软件工程）
 - 问题求解方法(4/15)
 - 枚举、模拟、分治；字符串、高精度数
 - 编程实践示例(5/15)
 - 物理随机模拟、统计数据分析、机器学习应用
 - 编程工程环境(3/15)
 - 高性能计算/并行计算/超级计算



课程信息

- 参考教材有两本，分别是MIT和Princeton的编程课程教材。有中文翻译版、电子版。
 - Introduction to Computation and Programming Using Python, 2nd Edition, by John V. Guttag, MIT Press, 2013
 - Introduction to Programming in Python: An Interdisciplinary Approach, by Robert Sedgewick, Kevin Wayne and Robert Dondero, Addison-Wesley Professional, 2015



课程信息

- 成绩评定
 - 论文或报告（个人完成）
 - 最终截止时间：预计在**2021年7月**
 - 过程管理
 - 2021年5月提交第一版（选题）
 - 2021年7月提交第二版（求解）



提纲

- 计算思维：原理、方法、工具
- 软件安装：Python, IDE (Anaconda)
- 需求帮助：提问的智慧
- 蒙特卡罗模拟
 - Pascal问题
 - 计算圆周率

Monte Carlo method

(蒙特卡罗模拟)



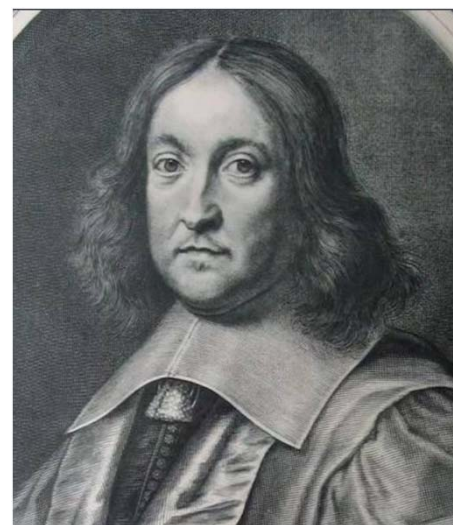
Stan Ulam, Feynman and von Neumann

Pascal问题

“连续掷一对骰子24次得到两个6”，是否可以下注？



Blaise Pascal (1623–1662)



Pierre de Fermat (1601–1665)

Pascal问题

“连续掷一对骰子24次得到两个6”，是否可以下注？

$$1 - (35/36) ** 24$$

```
#Figure 16.1
def rollDie():
    return random.choice([1, 2, 3, 4, 5, 6])

def checkPascal(numTrials):
    """Assumes numTrials an int > 0
    Prints an estimate of the probability of winning"""
    numWins = 0
    for i in range(numTrials):
        for j in range(24):
            d1 = rollDie()
            d2 = rollDie()
            if d1 == 6 and d2 == 6:
                numWins += 1
                break
    print('Probability of winning =', numWins/numTrials)
```



Blaise Pascal (1623–1662)



Pierre de Fermat (1601–1665)



双骰子（Double Dice）博弈

- Craps游戏用的骰子就是我们常见的六面骰子，玩家丢出两个骰子，骰子滚动停止之后，朝上的点数相加之后，决定输赢。
 - 过线投注：投出7或11，赢；投出2，3，12，输；投出其他点（4，5，6，8，9，10），继续投，投出与第一次相同的点赢，如先投出7则输。
 - 不过线：投出2或3，赢；投出7或11，输；投出12则平均；投出其他点（4，5，6，8，9，10），继续投，先投出与第一次相同的点则输，如先投出7则赢。

```

#figure 16.2
class CrapsGame(object):
    def __init__(self):
        self.passWins, self.passLosses = 0, 0
        self.dpWins, self.dpLosses, self.dpPushes = 0, 0, 0

    def playHand(self):
        throw = rollDie() + rollDie()
        if throw == 7 or throw == 11:
            self.passWins += 1
            self.dpLosses += 1
        elif throw == 2 or throw == 3 or throw == 12:
            self.passLosses += 1
            if throw == 12:
                self.dpPushes += 1
            else:
                self.dpWins += 1
        else:
            point = throw
            while True:
                throw = rollDie() + rollDie()
                if throw == point:
                    self.passWins += 1
                    self.dpLosses += 1
                    break
                elif throw == 7:
                    self.passLosses += 1
                    self.dpWins += 1
                    break

    def passResults(self):
        return (self.passWins, self.passLosses)

    def dpResults(self):
        return (self.dpWins, self.dpLosses, self.dpPushes)

```





#Figure 16.3

```
def crapsSim(handsPerGame, numGames):  
    """Assumes handsPerGame and numGames are ints > 0  
    Play numGames games of handsPerGame hands; print results"""  
    games = []  
  
    #Play numGames games  
    for t in range(numGames):  
        c = CrapsGame()  
        for i in range(handsPerGame):  
            c.playHand()  
            games.append(c)  
  
    #Produce statistics for each game  
    pROIperGame, dpROIperGame = [], []  
    for g in games:  
        wins, losses = g.passResults()  
        pROIperGame.append((wins - losses)/float(handsPerGame))  
        wins, losses, pushes = g.dpResults()  
        dpROIperGame.append((wins - losses)/float(handsPerGame))  
  
    #Produce and print summary statistics  
    meanROI = str(round((100*sum(pROIperGame)/numGames), 4)) + '%'  
    sigma = str(round(100*stdDev(pROIperGame), 4)) + '%'  
    print('Pass:', 'Mean ROI =', meanROI, 'Std. Dev. =', sigma)  
    meanROI = str(round((100*sum(dpROIperGame)/numGames), 4)) + '%'  
    sigma = str(round(100*stdDev(dpROIperGame), 4)) + '%'  
    print('Don\'t pass:', 'Mean ROI =', meanROI, 'Std Dev =', sigma)  
  
crapsSim(1000000, 10)
```




```
#Figure 16.4
def playHand(self):
    #An alternative, faster, implementation of playHand
    pointsDict = {4:1/3, 5:2/5, 6:5/11, 8:5/11, 9:2/5, 10:1/3}
    throw = rollDie() + rollDie()
    if throw == 7 or throw == 11:
        self.passWins += 1
        self.dpLosses += 1
    elif throw == 2 or throw == 3 or throw == 12:
        self.passLosses += 1
        if throw == 12:
            self.dpPushes += 1
        else:
            self.dpWins += 1
    else:
        if random.random() <= pointsDict[throw]: # point before 7
            self.passWins += 1
            self.dpLosses += 1
        else: # 7 before point
            self.passLosses += 1
            self.dpWins += 1
```



计算圆周率

- 阿基米德（前287—前212年）
 - 96边形， $223/71$ — $22/7$ ，3.1418
- 刘徽（约225年—约295年）
 - 3.1416
- 祖冲之（429年—500年）
 - 3.1415926和3.1415927之间
 - 16世纪才有更精确的结果
- 布冯（1707—1788）
- 拉普拉斯（1749—1827）





计算圆周率

```
#Figure 16.6
def throwNeedles(numNeedles):
    inCircle = 0
    for Needles in range(1, numNeedles + 1):
        x = random.random()
        y = random.random()
        if (x*x + y*y)**0.5 <= 1:
            inCircle += 1
    #Counting needles in one quadrant only, so multiply by 4
    return 3*(inCircle/numNeedles)

def getEst(numNeedles, numTrials):
    estimates = []
    for t in range(numTrials):
        piGuess = throwNeedles(numNeedles)
        estimates.append(piGuess)
    sDev = stdDev(estimates)
    curEst = sum(estimates)/len(estimates)
    print('Est. =', str(round(curEst, 5)) + ', ',
          'Std. dev. =', str(round(sDev, 5)) + ', ',
          'Needles =', numNeedles)
    return (curEst, sDev)

def estPi(precision, numTrials):
    numNeedles = 1000
    sDev = precision
    while sDev > precision/1.96:
        curEst, sDev = getEst(numNeedles, numTrials)
        numNeedles *= 2
    return curEst
```

