**Python实验报告9**

**学号：** 117060400130 **姓名**： 罗建燕 **班级：** 17应用统计（1）班  **指导老师：** 林卫中

**实验1**：

**算法实现：from random import randomdef printIntro(): print("这个程序模拟两个选手A和B的某种竞技比赛") print("程序运行需要A和B的能力值（以0到1之间的小数表示）")def getInputs(): a = eval(input("请输入选手A的能力值(0-1): ")) b = eval(input("请输入选手B的能力值(0-1): ")) n = eval(input("模拟比赛的场次: ")) return a, b, ndef simNGames(n, probA, probB): winsA, winsB = 0, 0 for i in range(n): scoreA, scoreB = simOneGame(probA, probB) if scoreA > scoreB: winsA += 1 else: winsB += 1 return winsA, winsBdef gameOver(a,b): return a==21 or b==11def simOneGame(probA, probB): scoreA, scoreB = 0, 0 serving = 0 t=0 while not gameOver(scoreA, scoreB): if serving == 0: if random() < probA: scoreA += 1 else: scoreB +=1 # serving="B" else: if random() < probB: scoreB += 1 else: scoreA += 1 t+=1 if t%5==0: serving=(serving+1)%2 #serving="A" return scoreA, scoreBdef printSummary(winsA, winsB): n = winsA + winsB print("竞技分析开始，共模拟{}场比赛".format(n)) print("选手A获胜{}场比赛，占比{:0.1%}".format(winsA, winsA/n)) print("选手B获胜{}场比赛，占比{:0.1%}".format(winsB, winsB/n))def main(): printIntro() probA, probB, n = getInputs() winsA, winsB = simNGames(n, probA, probB) printSummary(winsA, winsB)main()实验2：**

**算法实现：from random import randomdef printIntro(): print("这个程序模拟两个选手A和B的某种竞技比赛") print("程序运行需要A和B的能力值（以0到1之间的小数表示）")def getInputs(): a = eval(input("请输入选手A的能力值(0-1): ")) b = eval(input("请输入选手B的能力值(0-1): ")) n = eval(input("模拟比赛的场次: ")) return a, b, ndef simNGames(n, probA, probB): winsA, winsB = 0, 0 for i in range(n): scoreA, scoreB = simOneGame(probA, probB) if scoreA > scoreB: winsA += 1 else: winsB += 1 return winsA, winsBdef gameOver(a,b): """ if (a<10 and b==11) or (a==11 and b<10): return True if(a>=10 and b-a==2) or(b>=10 and a-b==2): return True return False""" if a<b: a,b=b,a if a==11 and b<10: return True if b>=10 and a-b==2: return True return Falsedef simOneGame(probA, probB): scoreA, scoreB = 0, 0 serving = 0 t=0 while not gameOver(scoreA, scoreB): if serving == 0: if random() < probA: scoreA += 1 else: scoreB +=1 # serving="B" else: if random() < probB: scoreB += 1 else: scoreA += 1 t+=1 if t%2==0: serving=(serving+1)%2 #serving="A" return scoreA, scoreBdef printSummary(winsA, winsB): n = winsA + winsB print("竞技分析开始，共模拟{}场比赛".format(n)) print("选手A获胜{}场比赛，占比{:0.1%}".format(winsA, winsA/n)) print("选手B获胜{}场比赛，占比{:0.1%}".format(winsB, winsB/n))def main(): printIntro() probA, probB, n = getInputs() winsA, winsB = simNGames(n, probA, probB) printSummary(winsA, winsB)main()实验3：**

**算法实现：from random import randomfrom random import randintdef printIntro(): print("这个程序模拟两个选手A和B的某种竞技比赛") print("程序运行需要A和B的能力值（以0到1之间的小数表示）")def getInputs(): a = eval(input("请输入选手A的能力值(0-1): ")) b = eval(input("请输入选手B的能力值(0-1): ")) n = eval(input("模拟比赛的场次: ")) g1=eval(input("A投篮能力:")) g2=eval(input("B投篮能力:")) return a, b, n,g1,g2def simNGames(n, goleA, boardA,goleB,boardB): winsA, winsB = 0, 0 for i in range(n): scoreA, scoreB = simOneGame(goleA, boardA,goleB,boardB) if scoreA > scoreB: winsA += 1 else: winsB += 1 return winsA, winsBdef gameOver(t): return t>=12\*60def simOneGame(goleA, boardA,goleB,boardB): scoreA, scoreB = 0, 0 serving = 0 totalTime=0 while not gameOver(totalTime): t=randint(1,24) totalTime += t if t == 24: serving=(serving + 1)%2 else: if serving == 0: if random() < goleA: scoreA += 1 serving=1 else: if random()<boardA: serving=0 else: serving=1 else: if random() < goleB: scoreB += 1 serving=0 else: if random()<boardB: serving=1 else: serving=0 return scoreA, scoreBdef printSummary(winsA, winsB): n = winsA + winsB print("竞技分析开始，共模拟{}场比赛".format(n)) print("选手A获胜{}场比赛，占比{:0.1%}".format(winsA, winsA/n)) print("选手B获胜{}场比赛，占比{:0.1%}".format(winsB, winsB/n))def main(): printIntro() goleA, boardA,goleB,boardB, n = getInputs() winsA, winsB = simNGames(n,goleA, boardA,goleB,boardB) printSummary(winsA, winsB)main()**

**实验四：**

**算法实现:** **import jiebaimport matplotlib.pyplot as pltfrom wordcloud import WordCloud, ImageColorGeneratorimport numpy as npimport PIL.Image as Imagedef calWordFrequence(): excludes = {} # {"将军","却说","丞相"} txt = open("三国演义.txt", "r", encoding='utf-8').read() words = jieba.lcut(txt) counts = {} for word in words: if len(word) == 1: continue else: counts[word] = counts.get(word, 0) + 1 for word in excludes: del (counts[word]) return countsdef drawWordCloud(counts): coloring = np.array(Image.open("E:/baidupic/9.png")) wc = WordCloud(background\_color="white", max\_words=2000, mask=coloring, max\_font\_size=60, random\_state=42, scale=2, font\_path="c:/Windows/Fonts/SimHei.ttf") wc.generate\_from\_frequencies(counts) image\_colors = ImageColorGenerator(coloring) plt.imshow(wc) plt.axis("off") plt.show()def main(): counts = calWordFrequence() drawWordCloud(counts)main()**