**《最优化技术》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | | **2019级计算机科学与技术（卓越）02** | | | **姓名** | **李燕琴** |
| **实验题目** | 动态规划算法的应用 | | | | | |
| **实验时间** | **2021年5月21日** | | **实验地点** | **DS3401** | | |
| **实验成绩** |  | | **实验性质** | **□验证性 ■设计性 □综合性** | | |
| 教师评价：  □算法/实验过程正确；□源程序/实验内容提交 □程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； □报告规范；  其他：  评价教师签名： | | | | | | |
| 一、实验目的  理解动态规划算法的基本思想，并应用于求解实际问题。 | | | | | | |
| 二、实验项目内容  请利用动态规划解决以下两个问题：   1. 爬楼梯问题。自行输入n阶楼梯，每次可以爬1或2个台阶，请编程判断有多少种不同方法可以爬完楼梯。 2. 判定买卖股票的最佳时机。自行给定一个数组，它的第i个元素是一支给定股票第i天价格。设计一个算法来计算你所能获取的最大利润。注意你不能在买入股票前卖出股票，不能同一天进行多次交易。   例如：  输入： [7,1,3,5,4,6]  输出：最大获利：6  对以上结果的解释：第二天买入股票，第四天卖出股票，则获利为5-1=4元，第五天买入股票，第六天卖出股票，获得2元，总共获得6元。  注意：所有程序请用python语言实现。只提交本电子文档，注意本文件末尾的文件命名要求；源程序一节请用代码备注的方式说明你的算法和思路；实验结果一节需要提供测试结果截图并给出结果分析。 | | | | | | |
| 三、实验过程或算法（源程序）  1）爬楼梯  import numpy as np  n = int(input("请输入楼梯层数 n = "))  dp = np.zeros(n+1)  dp[1] = 1  dp[2] = 2  # 思路陈述：dp[i]表示第i阶层有dp[i]种方法，  # 其中可以源于第i-1层走一步，也可以源于第i-2层走2步，  # 故得到动态规划转移方程 dp[i] = dp[i-1]+dp[i-2]  for i in range(3,n+1):      dp[i] = dp[i-1]+dp[i-2]  print("爬楼梯的方法有%d种"%(dp[n]))  2）买股票  import re  s = input()  # 格式化输入  sp = re.split(r',|\[|\]',s)  # 转为数字  dp = [int(sp[i]) for i in range(1,len(sp)-1)]  res = 0  # 思路陈述：  # 若股票价格连续上涨有p1, p2, …, pn，则第一天买最后一天卖收益最大，  # 即pn - p1，等价于每天都买卖pn - p1 = (p2 - p1) + (p3 - p2) + … + (pn - pn-1)；  # 若股票价格连续下降，则不做买卖，收益最大（不会亏钱）。  for i in range(1,len(dp)):      if dp[i]>dp[i-1]:          res = res + dp[i] - dp[i-1]  print("最大获利：%d"%(res)) | | | | | | |
| 四、实验结果及分析和（或）源程序调试过程  1）爬楼梯  测试符合预期。        2）买股票  测试符合预期。 | | | | | | |

注：电子文档命名要求：学号+姓名+实验序号