**《最优化技术》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | | **2020级计算机科学与技术01班** | | | **姓名** | **陈鹏宇** |
| **实验题目** | 动态规划算法的应用 | | | | | |
| **实验时间** | **2022年 04月 25日** | | **实验地点** | **Ds1421** | | |
| **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。   1. 有一个国家发现了5座金矿，每座金矿的黄金储量不同，需参与挖掘的工人数也不同，具体情况如下表所示。参与挖矿的工人总数为10人，每座矿要么全挖要么不挖。要求用程序求解出，要想得到尽可能多的黄金，应选择挖取哪几座金矿？  |  |  |  |  |  | | --- | --- | --- | --- | --- | | No.1 | No.2 | No.3 | No.4 | No.2 | | 200金/3人 | 300金/4人 | 350金/3人 | 400金/5人 | 500金/5人 |   注意：实验语言不限，只提交本电子文档，注意本文件末尾的文件命名要求；源程序一节请先给出你的算法思路，并用代码备注的方式说明算法的关键点；实验结果一节需要提供调试过程中的中间结果及测试最终结果的截图并给出结果分析，若在调试过程中有错，请给出错误原因分析。 | | | | | | |
| 1. 实验过程或算法（源程序）   1.楼梯问题算法思路  采用动态规划从底向上的方法，根据状态转移方程和初始状态  可以得到到达每一层楼梯的不同方法这一重叠子问题，最后输出及为爬完n层楼梯的方法数。  楼梯问题代码(c++)  #include <iostream>  #include <bits/stdc++.h>  using namespace std;  int main(){  int n ; cin >> n; //输入楼梯总数  int dp[n+1]; //动态规划表记录每一层楼梯的方法数  dp[0] = 0 ; dp[1] = 1 ; dp[2] = 2;//初始化  for(int i = 3 ; i <= n ; i++){ //从底向上循环求解  dp[i] = dp[i-1] + dp[i-2]; //每一次只能上1-2层  }  cout << dp[n] << " ways in total" << endl; //输出结果  return 0;  }  2.股票问题算法思路  考虑到每天只能进行一次交易，且不能在购入股票前卖出股票，所以我们将分别考虑当天是否持有股票。dp[0][i]表示第i天持有股票的情况下的最大收益，dp[1][j]表示第i天未持有股票的情况下的最大收益。状态转移方程和初始状态为  第i天持有股票的情况要么是当天才买，要么是前一天没卖  第i天未持股的情况要么是当天才卖，要么是前一天也未持股  由题，未持股的最后一天即为本次股票买卖的最大收益。  股票问题代码  #include <iostream>  #include <bits/stdc++.h>  using namespace std;  int main(){  vector<int> v;  char junk ;  while(cin >> junk){ //解决输入格式问题  if(junk != '[' && junk != ']' && junk != ','){  int tmp = junk-'0';  v.push\_back(tmp);  }  if(junk == ']')  break;  }  int n = v.size();  int dp[2][n+1]; //dp表得到每一天最大收益情况，第一行代表当天持有股票，第二行代表当天不持有股票  dp[0][1] = -v[0]; //第一天持有股票的情况下，利润为负  dp[1][1] = 0; //第一天不持有股票，利润为0  for(int i = 2 ; i <= n ; i++){  dp[0][i] = max(dp[1][i-1] - v[i-1] , dp[0][i-1]);//第i天持有股票，说明第i天购入股票或者和第i-1天一样  dp[1][i] = max(dp[0][i-1] + v[i-1] , dp[1][i-1]);//第i天不持有股票，说明第i天卖出股票或者第i天和第i-1天一样  }  for(int i = 1 ; i <= n ; i++)  cout<<dp[0][i]<<" "<<dp[1][i]<<endl;  cout<<"max profit is: "<<dp[1][n]<<endl; //最后一天肯定不会持股  return 0;  }  3.挖矿问题算法思路  挖矿问题与背包问题相似，我们可以把每一座矿山看作一个物品，其中矿山的金子数为物品的价值，矿山的人力资源消耗为矿山的代价，有限的人力资源可以看为有限的背包容量，所以求解思路与01背包问题一致，我们得到如下状态转移方程和初始状态  dp[n][m]即为最大黄金收益。  而要得到挖了哪些金矿，只需从后往前递推即可  如果，则说明挖了第n座金矿。  挖矿问题代码  #include <iostream>  #include <bits/stdc++.h>  using namespace std;  int weight[5] = {3,4,3,5,5}; //每座金矿需要的人力资源  int value[5] = {200,300,350,400,500}; //每座金矿的价值  int bag = 10; //一共10个人  int dp[6][11]; //dp table  int main(){  for(int i = 0 ; i < 11 ; i++){ //initialize  dp[i][0] = 0;  dp[0][i] = 0;  }  for(int i = 1 ; i < 6 ; i++){  for(int j = 1 ; j < 11 ; j++){  if(j < weight[i-1]) //如果当前出动的人力无法采集第i-1座矿  dp[i][j] = dp[i-1][j];  else{  dp[i][j] = max(dp[i-1][j-weight[i-1]] + value[i-1] , dp[i-1][j]); //采集当前矿和不采集的最大值  }  }  }  for(int i = 0 ; i < 6 ; i++){  for(int j = 0 ; j < 11 ; j++){  cout<<dp[i][j]<<" ";  }  cout<<endl;  }  //现开始查找挖了哪些矿  int i = 5; int j = 10;  int tmp = dp[5][10]; //从后往前递推得到挖矿的具体情况  while(tmp){  if(tmp - value[i-1] == dp[i-1][j-weight[i-1]]){  cout<<"item " << i << " mine is dug" <<endl;  tmp -= value[i-1];  j -= weight[i-1];  i --;  }  else  i--;  }  cout<<"the max value is "<<dp[5][10]<<endl;  } | | | | | | |
| 1. 实验结果及分析和（或）源程序调试过程   1.楼梯问题实验结果  IMG_256IMG_256  n=3时，共有dp[2] + dp[1]即3种方法，具体为1+1+1，1+2，2+1  n=4时，共有dp[3] + dp[2]即5种方法，具体为1+1+1+1，1+2+1，2+1+1，2+2，1+1+2  n=5时，共有dp[4] + dp[3]即8种方法，  具体方法为n=3的方法后面+2和n=4的方法后面+1  以此类推  n=10时，共有dp[9] + dp[8]即89种方法  楼梯问题程序求解过程中，一开始由于没有把n=2初始化，导致计算n=2时  dp[2] = dp[0] + dp[1] = 1  而出错，将dp[2]初始化为2后，后续结果正确。  2.股票问题实验结果  IMG_256  实验结果正确，现输出中间结果详细分析  IMG_256  第一天-7，0为初始状态，  第二天在第二天购入股票比保留第一天股票利润更高，所以选择在第二天购入  在第二天出售第一天股票的利润不如不买股票，所以第二天与第一天保持一致  第三天持有股票与第二天一致  卖出股票后赚2  第四天持有股票与第三天一致  但卖出股票比第三天卖出收益更高，所以更新收益  第五天购买股票比保留股票收益更高  卖出股票没有第四天卖出股票收益高  第六天保留股票比购买股票收益更高  卖出股票比第五天卖出股票收益高，最终收益6  3.挖矿问题实验结果  IMG_256  如上图，想要得到最大的黄金收益，需要挖第五座和第四座矿山，最大收益为900.  我们输出这个6行11列的dp表进行分析  当开放第一座矿山且人力资源(简称j)大于等于3时，最大收益(简称a)为200  当开放第二座矿山时，j=3时a=200，但j=4时，可以开采第二座矿山，a=300  J=7时两座矿山可以同时开采，a=500  以此类推，最终得到900的收益。  查找矿山开采状态时，从右下角开始，900-500=400=dp[5][5]，所以第五座矿山开采了  400-400=0=dp[4][0]，所以第四座矿山开采了。 | | | | | | |