**《算法设计与分析》作业**

# 作业二 动态规划

|  |  |  |  |
| --- | --- | --- | --- |
| **作业** | | | |
| 姓名 | 吴宇敖 | 指导教师 |  |
| 学号 | 176001752 | 日 期 |  |
| 班级 | 17计算机软件一班 |  |  |
| **作业内容** | | | |
| **[要求先在OJ上提交通过]**   1. 预习教材并查找资料，学习动态规划基本知识。 2. 学习教材例题：最大连续字段和；完成OJ 3. 学习教材例题：01背包：完成OJ[减肥的小K2]；提示：采或不采 | | | |
| **作业解答** | | | |
| **【实验题目】**  **小K是个苦命的孩子，他的师傅为了多赚钱，以减肥为理由，让他去采药，并说不完成不能吃饭。野地里有许多不同的草药，采每一株都需要一些时间，每一株也有它自身的价值。要求在规定的时间t里，采到的草药的总价值最大。**  **输入要求：**  **第一行有2个整数T(1≤T≤1000)和M(1≤M≤100)，一个空格隔开，T代表总共能够用来采药的时间，M代表山洞里的草药的数目。**  **接下来的M行每行包括两个在1到100之间包括1和100）的整数，分别表示采摘某株草药的时间和这株草药的价值。**  **输出要求：**  **1个整数，表示在规定的时间内可以采到的草药的最大总价值。**  **【程序代码及注释】**  #include<iostream>  using namespace std;  int main()  {  int T,M;//一共可以采药的时间和药材的数目  cin>>T>>M;  int time[100];//每个药材需要的时间  int price[100];//每个药材的价钱  for(int i=1;i<=M;i++)//初始化ˉ  cin>>time[i]>>price[i];  int res[101][1001]={0};  for(int i=1;i<=M;i++)  for(int j=1;j<=T;j++)  if(j>=time[i])//当时间大于等于采药时间，比较不采这一株药的价值与采了这一株药的价值  res[i][j]=max(res[i-1][j],res[i-1][j-time[i]]+price[i]);  else  res[i][j]=res[i-1][j];  cout<<res[M][T]<<endl;  //system("pause");  return 0;  }  **【实验结果及遇到的问题】（附输入输出）**    问题：在两个价值比较选最大时，一开始，没有完全整理清晰思路，导致输出的结果变得非常大。  **【算法分析】**  这次实验是通过两个循环来计算，药材采摘的最大价值。第N株的最大价值应为：（1）当时间够时，应比较第N-1株发费相同时间的价值和第N-1株时花费时间减去第N株时间的价值再加上第N株的价值。（2）当时间不够时，应该直接与第N-1株发费相同时间的价值相同。  **【实验题目】**  **给出长度为n的数组，求最大连续子段和， 输出该最大和。**  **输入要求：**  **第1行输入一个整数n<50；表示输入数组的大小**  **第2行输入n个数，中间用空格隔开**  **输出要求：**  **最大连续子段和**  **【程序代码及注释】**  #include<iostream>  using namespace std;  int main()  {  int n,m;  cin>>n;  int num[50];  int res[50]={0};  for(int i=0;i<n;i++)//初始化  {  cin>>num[i];  }  res[0]=num[0];//res[0]的最大段的值为num[0]  for(int i=1;i<n;i++)//res[i]的最大段应该为（res[i-1]最大段的值与第n个数相加的结果）与第n个数的最大值  res[i]=max(num[i],res[i-1]+num[i]);  m=res[0];  for(int i=1;i<n;i++)//寻找0到n之间每个的段的最大值，取出最大值  {  if(m<res[i])  m=res[i];  }  cout<<m<<endl;  //system("pause");  return 0;  }  **【实验结果及遇到的问题】（附输入输出）**    问题：res[50]一开始忘记初始化，导致结果为-8.......，寻找最大值时，m直接定义为-1，导致当全是负数输入时，输出的结果为-1。  **【算法分析】**  这次实验为计算最大段的和，只需通过一个循环，将res[0]的值定义为第一个数的值，然后第N个段的和的最大值，应该为第N个数与第N-1个段的和的最大值加上第N个数的最大的那一个。最后只需要寻找到全部段的和的最大值中最大的那一个。 | | | |