**《算法设计与分析》实验报告**

# 实验二 动态规划

|  |  |  |  |
| --- | --- | --- | --- |
| **报告书** | | | |
| 姓名 | 吴宇敖 | 指导教师 |  |
| 学号 | 176001752 | 日 期 | 2020.4.11 |
| 班级 | 17计算机软件一班 |  |  |
| **实验内容** | | | |
| **[要求先在OJ上提交通过]**  1)在一个数字序列中，找到一个最长的非连续子序列，使得这个子序列是不下降（非递减）。现有序列A={1,2,3,-1,-2,7,9},则A的最长不下降子序列是{1,2,3,7,9}。如果有多个最长序列，只需选数字顺位靠后的序列从大到小输出。[OJ]  2)一条街上有N个商铺；商铺i有价值V[i]的物品，你有足够的时间在晚上光顾所有的商店，人们称呼你为盗贼；每个商店都有一个报警器，会在晚上报警，但是只有相邻的2个商店同时报警时，警察才会出动；你需要证明你是个合格的盗贼。[OJ] | | | |
| **实验目的** | | | |
| 1）练习使用动态规划 | | | |
| **实验过程和步骤** | | | |
| **以下仅供参考，请在报告中删除**  **【实验题目】**  在一个数字序列中，找到一个最长的非连续子序列，使得这个子序列是不下降（非递减）。现有序列A={1,2,3,-1,-2,7,9},则A的最长不下降子序列是{1,2,3,7,9}。如果有多个最长序列，只需选数字顺位靠后的序列从大到小输出。  **【程序代码及注释】**  #include<iostream>  using namespace std;  int main()  {  int n,m,num[1001],res[1001]={0};  cin>>n;  for(int i=0;i<n;i++)//输入  cin>>num[i];  int len[1001];//len是记录1到n 的最大的数列的长度  for(int i=0;i<n;i++)//初始化ˉ  len[i]=1;  for(int i=1;i<n;i++)  for(int j=0;j<i;j++)  if(num[i]>=num[j]&&len[i]<=len[j]+1)  {  len[i]=len[j]+1;  res[i]=j;//标记每个最大数列的数的前一个数在什么地方  }  for(int i=1,k=len[0];i<n;i++)//寻找最长的数列  if(k<=len[i])  {  k=len[i];  m=i;  }  cout<<len[m]<<endl;//输出长度  for(int i=m;i>0;)//输出数列  {  cout<<num[i]<<' ';  i=res[i];  if(len[i]==1)  {  cout<<num[i];  break;  }  }  cout<<endl;  //system("pause");  return 0;  }  **【实验题目】**  一条街上有N个商铺；商铺i有价值V[i]的物品，你有足够的时间在晚上光顾所有的商店，人们称呼你为盗贼；每个商店都有一个报警器，会在晚上报警，但是只有相邻的2个商店同时报警时，警察才会出动；你需要证明你是个合格的盗贼。  **【程序代码及注释】**  #include<iostream>  using namespace std;  int main()  {  int n,p[101],res[101]={0};//初始化  cin>>n;  for(int i=0;i<n;i++)//输入  cin>>p[i];  res[0]=p[0];  res[1]=max(p[0],p[1]);//先选择1家店铺和两家店铺的情况  for(int i=2;i<n;i++)//根据前两种来讨论之后的情况  res[i]=max(res[i-1],res[i-2]+p[i]);  cout<<res[n-1]<<endl;  //system("pause");  return 0;  } | | | |