**中国矿业大学计算机学院实验报告**

|  |
| --- |
| 课程名称  算法导论           实验名称 算法导论实验二  班级 计科17-6 姓名 郭子杨 学号 08173042  仪器组号\_\_\_\_\_\_\_  实验日期 10.22  实验报告要求：1.实验目的   2.实验内容（题目描述，源代码，运行截图，调试情况）    3.实验体会 |
| 1. 实验目的   1.熟悉最长公共子序列问题的算法；  2.初步掌握动态规划算法；  二、实验内容  1、第一题  1.1题目描述  【问题描述】  给定n种物品和一个背包，物品i的重量是Wi，其价值为Vi，问如何选择装入背包的物品，使得装入容量为c的背包的物品的总价值最大？  【输入形式】    第一行输入背包容量c；第二行输入要装入的物品的个数；第三行输入装入的各个物品的重量w[i]；第四行输入装入的物品的价值v[i]          i 【输出形式】输出背包的最大总价值 【样例输入】  6  5  3 2 1 4 5  25 20 15 40 50  1.2 源代码  #include<iostream>  using namespace std;  int main()  {  int c,n;  cin>>c>>n;  int w[25]={0},v[25]={0};  int sum[25][25]={0};  for(int i=1;i<=n;i++)  {  cin>>w[i];  }  for(int j=1;j<=n;j++)  {  cin>>v[j];  }  for(int k=1;k<=n;k++)  for(int p=c;p>=0;p--)  {  if(p>=w[k])  {  if(sum[k-1][p-w[k]]+v[k]>=sum[k-1][p])  {  sum[k][p]=sum[k-1][p-w[k]]+v[k];  }    else  {  sum[k][p]=sum[k-1][p];  }  }  else  sum[k][p]=sum[k-1][p];  }  cout<<sum[n][c]<<endl;  return 0;  }  1.3 运行截图    1.4 调试情况  无  2、第二题  2.1 题目描述  【问题描述】  若给定n个整数组成的序列a1，a2，a3，……an，求该序列形如ai＋ai＋1＋……＋an的最大值。  注意：当所有整数均为负整数时定义此时最大子段和为0.  【输入形式】第一行输入所输入数的个数，第二行输入各个数 【输出形式】该数组的最大子段和 【样例输入】  7  -1 -3 -4 5 7 -8 -9  2.2 源代码  #include<iostream>  using namespace std;  int sequence(int a[],int n)  {  int max=0;  int b[200];  if(a[0]>0)  {  b[0]=a[0];  }  else  {  b[0]=0;  }  for(int j=1;j<n;j++)  {  if(b[j-1]+a[j]>=a[j])  {  b[j]=b[j-1]+a[j];  }  else  {  b[j]=a[j];  }  if(max<b[j])  {  max=b[j];    }  }  return max;  }  int main()  {  int n;  cin>>n;  int a[200];  for(int i=0;i<n;i++)  {  cin>>a[i];  }  int k=sequence(a,n);  cout<<k<<endl;  return 0;  }  2.3 运行截图    2.4 调试情况  无  3.第三题  3.1 题目描述  【问题描述】  若给定序列X={x1,x2,…,xm}，则另一序列Z={z1,z2,…,zk}，是X的子序列是指存在一个严格递增下标序列{i1,i2,…,ik}使得对于所有j=1,2,…,k有：zj=xij。例如，序列Z={B，C，D，B}是序列X={A，B，C，B，D，A，B}的子序列，相应的递增下标序列为{2，3，5，7}。  给定2个序列X和Y，当另一序列Z既是X的子序列又是Y的子序列时，称Z是序列X和Y的公共子序列。  给定2个序列X={x1,x2,…,xm}和Y={y1,y2,…,yn}，找出X和Y的最长公共子序列。  【输入形式】  第一行为一个整数，表示序列X的字符数  第二行依次输入序列X的字符  第三行为一个整数，表示序列Y的字符数  第四行依次输入序列Y的字符  【输出形式】  输出X和Y的最长公共子序列 【样例输入】  7  ABCBDAB  6  BDCABA 【样例输出】  BCBA  3.2 源代码  #include <algorithm>  #include <iostream>  #include <stdio.h>  #include <string>  #define MAXN 1111  #define INF 0x3f3f3f3f  using namespace std;  string str1;  string str2;  int dp[MAXN][MAXN];  int path[MAXN];  int main(void)  {  int n, m;  cin>>n;  cin >> str1;  cin>>m;  cin >> str2;  for (int i = 1; i <= n; i++)  for (int j = 1; j <= m; j++)  if (str1[i - 1] == str2[j - 1])  dp[i][j] = dp[i - 1][j - 1] + 1;  else  dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);  int x = 0;  for (int i = n, j = m;dp[i][j]!=0;)  {  if (dp[i][j] == dp[i - 1][j])  i--;  else if (dp[i][j] == dp[i][j - 1])  j--;  else  {  path[x++] = i - 1;  i--;  j--;  }  }  for (int i = x - 1; i >= 0; i--)  {  cout<<str1[path[i]];  }  cout<<endl;  return 0;  }  3.3 运行截图    3.4 调试情况  无  三、实验体会  通过本次上机学习，初步掌握动态规划算法，对课本上理论知识更加巩固，调试代码时也不免出现错误。 |