**算法分析动态规划project报告**

**姓名：邓露**

**学号：20184275**

**年级：2018级**

**专业班级：物联网工程专业1班**

**1、算法思想：**

使用动态规划的自底向上法求解最大总价值，对每个子问题只求解一次并将结果保存下来，数组value[i][j]保存第i排、第j列柱桩上的宝石价值，数组res[i][j] 是对子问题从第一排第一列到第i排、第j列可得的最大总价值,数组pos[i][j]保存第i排、第j列的最大价值来源于i-1排的列位置数。

则res[i][j]=value[i][j]+max{res[i-1][j-1],res[i-1][j],res[i-1][j+1]}。

伪代码为：

Dynamic-Button-Up (m, n, value):

let res[0..m][0..n],pos[0..m][0..n] be a new array

pos[0][1]=1

for i=0 to m+1

res[i][j]=0

for i=1 to m

for j=1 to i+1

res[i][j]=value[i][j]+max{res[i-1][j-1],res[i-1][j],res[i-1][j+1]}

pos[i][j]=j //max{res[i-1][j-1],res[i-1][j],res[i-1][j+1]}最大值对//应的列位置。

**2、递推方程式：**

公式为:

res[i][j]=

i取值从1到m, j取值为1到i。

**3、最优子结构性质：**

对共i排j列的最大总价值问题，假设最大总价值在i排k列，则在i-1排j列的最大总价值问题中，需要求解i-1排k-1列、i-1排k列和i-1排k+1列的最大总价值。

则i排k列的最优解包含其子问题（i-1排k-1列、i-1排k列和i-1排k+1列）的最优解，则该最大总价值问题具有最优子结构性质。

**4、程序结构描述：**

(a).（主要程序project.cpp）

首先进行读文件操作，读入test.txt文件数据，并存入value[][],初始化最大总价值res[][]、位置记录pos[][]。

其次是动态规划的主要部分，由下而上的计算每个子问题的最优解，res[i][j]=value[i][j]+max{res[i-1][j-1],res[i-1][j],res[i-1][j+1]}，并记录最优解来源位置，将得到max{res[i-1][j-1],res[i-1][j],res[i-1][j+1]}值的元素的列位置记入pos[i][j]。

最后是整理、输出、写文件。将最后一排每一个位置的最优解进行比较，得到该问题的最大总价值，并记录此位置到pos[m][1]。并由此回溯找到每一排的最大总价值的位置，并存入store[m]中。然后按照格式输出最大总价值和每一排的位置，并写入output.txt文件。

(b).（辅助程序newdata.cpp）

这段程序用于更新test.txt文件中的数据。

首先生成两个大于20（且小于50（project.cpp并没有限制小于50））的随机数m,n，对m排n列，生成m\*n个随机数（1~30）,输出并写入test.txt中

**5、时间复杂度分析：**

由上述算法思想中的伪代码可知，程序动态规划部分两个嵌套for循环，对i=1 to m,和对j=1 to i，时间复杂度为O(n^2)。程序后面整理、写文件部分，都是单个for循环，时间复杂度为O(n)。

所以，综上，时间复杂度为O(n^2)。

**6、测试结果：**

其中一组测试结果为：

/\*test.txt中的内容：\*/

20 30

21,18,4,22,28,27,1,30,6,17,8,13,18,6,18,19,5,19,25,2,9,18,22,19,16,10,29,14,15,10

20,13,27,3,14,2,6,28,1,19,28,30,22,27,11,6,14,13,7,8,12,28,8,13,3,3,25,6,22,5

24,16,12,5,14,21,17,8,12,13,7,17,18,30,4,5,6,28,6,30,28,13,2,6,13,16,8,9,27,8

3,21,14,29,24,25,20,3,25,27,24,27,2,19,30,9,13,14,15,8,7,23,28,26,12,5,16,11,21,13

1,27,27,14,8,21,13,5,15,2,3,1,24,29,25,27,10,24,16,10,16,5,4,20,22,10,14,5,21,15

3,11,18,28,19,29,20,29,1,1,15,3,19,11,28,26,12,25,6,16,17,20,22,24,7,25,29,11,16,12

20,20,30,30,22,30,1,14,30,25,12,9,7,29,8,17,4,1,18,20,11,28,5,14,22,13,19,10,16,28

5,12,17,19,27,30,3,21,4,18,15,30,23,23,11,28,21,7,8,29,5,28,8,1,4,7,10,29,25,6

21,10,10,29,26,29,9,13,4,26,25,10,5,17,3,5,23,23,28,27,2,3,6,1,9,25,11,17,14,20

25,11,26,10,8,15,23,4,24,27,19,20,18,27,30,24,25,13,5,23,14,3,10,6,4,27,6,1,17,21

7,8,20,30,26,7,20,22,18,24,13,2,18,6,18,26,9,30,11,14,14,21,28,15,3,7,29,10,5,5

18,23,13,22,22,22,22,3,20,14,10,27,23,28,26,27,28,13,30,5,25,24,2,12,30,15,20,30,26,13

23,16,27,4,14,3,10,17,27,21,6,1,20,17,14,17,17,27,30,1,27,14,13,15,5,27,3,14,20,12

2,1,10,12,14,6,7,2,23,1,5,25,27,28,28,25,11,4,3,24,19,22,15,27,11,7,6,23,24,18

1,24,27,22,6,16,8,20,6,14,23,8,30,26,18,17,4,3,17,23,19,3,3,26,29,3,3,14,14,17

21,29,29,11,1,7,8,23,14,8,3,1,27,14,12,18,19,18,19,10,24,24,29,28,9,27,27,3,11,23

23,20,26,27,13,30,17,30,7,19,14,12,21,23,2,27,11,4,3,25,17,14,20,9,8,30,12,17,23,16

5,5,29,12,4,10,15,19,18,19,3,13,5,20,26,1,8,27,29,9,29,2,26,3,12,14,22,6,6,6

29,29,23,14,5,10,22,30,21,19,5,22,4,29,20,26,19,9,12,5,8,29,1,15,30,24,23,7,25,1

4,28,25,17,28,25,25,25,21,15,12,6,17,17,25,10,1,23,30,28,26,3,18,21,16,10,20,14,28,14

/\*output.txt中的内容：\*/

513 （最大价值）

1 （开始位置，固定）

1 在第2排的位置

1 在第3排的位置

2 在第4排的位置

3 在第5排的位置

4 在第6排的位置

4 在第7排的位置

5 在第8排的位置

4 在第9排的位置

3 在第10排的位置

4 在第11排的位置

4 在第12排的位置

3 在第13排的位置

4 在第14排的位置

3 在第15排的位置

3 在第16排的位置

4 在第17排的位置

3 在第18排的位置

2 在第19排的位置

2 在第20排的位置

**7、源代码：**

（a）.project.cpp对test.txt中的数据进行计算并将结果存入output.txt中：

#include<iostream>

#include<stdio.h>

#include<stdlib.h>

#include<fstream>

using namespace std;

int Max(int a,int b){

return a>b?a:b;

}

int main(){

ifstream fin("test.txt");

if(!fin){

cout<<"读取文件失败！"<<endl;

return -1;

}

int m,n;//m排n列

fin>>m>>n;

char c;

int value[m+20][n+20]={0};

int res[m+20][n+20]={0};//每一个位置的最大值

int pos[m+20][n+20]={0};//记录最大值位置

for(int i=1;i<=m;i++){

for(int j=1;j<n;j++){

fin>>value[i][j]>>c;

res[i][j]=value[i][j];

}

fin>>value[i][n];

res[i][n]=value[i][n];

}

fin.close();

//动态规划

for(int i=1;i<m;i++){

for(int j=1;j<=i;j++){

if(j-1>0){

int temp=res[i+1][j-1];

res[i+1][j-1]=Max(temp,res[i][j]+value[i+1][j-1]);

if(res[i+1][j-1]!=temp){

pos[i][j-1]=j;

}

}

if(j+1<=n){

int temp=res[i+1][j+1];

res[i+1][j+1]=Max(temp,res[i][j]+value[i+1][j+1]);

if(res[i+1][j+1]!=temp){

pos[i][j+1]=j;

}

}

int temp=res[i+1][j];

res[i+1][j]=Max(temp,res[i][j]+value[i+1][j]);

if(res[i+1][j]!=temp){

pos[i][j]=j;

}

}

}

int maximun=res[m][1];

pos[m][1]=1;

for(int j=2;j<=n;j++){

int temp=res[m][j];

maximun=Max(temp,maximun);

if(maximun==temp){

pos[m][1]=j;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//test of res[][] and pos[][]

for(int i=1;i<=m;i++){

for(int j=1;j<=n;j++){

printf("%d ",res[i][j]);

}

cout<<endl;

}

for(int i=1;i<=m;i++){

for(int j=1;j<=n;j++){

printf("%d ",pos[i][j]);

}

cout<<endl;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

ofstream fout("output.txt",ios::trunc);

cout<<"测试结果"<<endl;

cout<<maximun<<"\t（最大价值）"<<endl;

fout<<maximun<<"\t（最大价值）"<<"\n";

int flag=1;

int store[m+20];

for(int i=m;i>0;i--){

store[i]=pos[i][flag];

flag=store[i];

}

cout<<store[1]<<"\t（开始位置，固定）"<<endl;

fout<<store[1]<<"\t（开始位置，固定）"<<"\n";

for(int i=2;i<=m;i++){

cout<<store[i]<<"\t在第"<<i<<"排的位置"<<endl;

fout<<store[i]<<"\t在第"<<i<<"排的位置"<<"\n";

}

fout.close();

cout<<"测试结果已更新到output.txt"<<endl;

system("pause");

getchar();

getchar();

return 0;

}

（b）.newdata.cpp生成随机数并以输入格式存入（更新）到test.txt中：

#include<iostream>

#include<cstdio>

#include<cstdlib>

#include<ctime>

#include<fstream>

using namespace std;

int main(){

srand((int)time(0));

int m,n;

m=rand()%30+20;

n=rand()%30+20;

ofstream fout("test.txt",ios::trunc);

cout<<"新的测试数据"<<endl;

cout<<m<<" "<<n<<endl;

fout<<m<<" "<<n<<"\n";

int k;

for(int i=0;i<m;i++){

for(int j=0;j<n-1;j++){

k=rand()%30+1;

fout<<k<<",";

cout<<k<<",";

}

k=rand()%30+1;

fout<<k<<"\n";

cout<<k<<endl;

}

fout.close();

cout<<"新的测试数据已更新到test.txt"<<endl;

system("pause");

return 0;

}