**东北大学信息科学与工程学院**

**数据结构课程设计报告**

题目 最佳任务分配方案

课题组长

课题组成员

专业名称 计算机科学与技术

班级 计1802

指导教师

2019 年 1月

**课程设计任务书**

|  |
| --- |
| **题目：**  **最佳任务分配方案** |
| **问题描述：**  设有n个人，准备承担m项课题任务，每个人只能承担其中的1项。一般情况下n≥m，可以有多种任务分配方案。假设每个人完成不同的课题所需的经费不同，则一定存在一种最佳方案，使得完成所有课题的总的费用最省。 |
| **设计要求：**  设计对于给定的任务分配方案，求最佳任务分配方案，使得费用最省。  （1）采用STL一维向量类构造二维向量矩阵。  （2）应用基本运算，通过搜索状态空间树实现问题求解。 |
| **指导教师签字：**  **年　　月　　日** |

1 课题概述

* 1. 课题任务

问题描述：设有n个人，准备承担m项课题任务，每个人只能承担其中的1项。一般情况下n≥m，可以有多种任务分配方案。假设每个人完成不同的课题所需的经费不同，则一定存在一种最佳方案，使得完成所有课题的总的费用最省。

设计要求：设计对于给定的任务分配方案，求最佳任务分配方案，使得费用最省。

（1）采用STL一维向量类构造二维向量矩阵。

（2）应用基本运算，通过搜索状态空间树实现问题求解。

* 1. 课题原理

课题解析：n个人和m项课题任务抽象成二分图，每个人对应多个课题任务，并具有各自的资金权值，题目要求求出完成所有课题且费用最省的方案。在n-m的二分图中，使用匈牙利算法计算方案的存在性并用KM算法求解。

匈牙利算法：通过不断寻找增广路径的办法，寻找最大路径。

KM算法：每次都帮一个顶点匹配最大权重边，利用匈牙利算法完成最大匹配，最终完成的就是最优匹配

* 1. 相关知识

1.3.1基本数据结构知识：向量的存储结构，匈牙利算法，KM算法的基本概念。

1.3.2 数据来源与存储：文件读取、键盘输入，c++vector。

1.3.3 c++的类及实现：采用c++类进行封装，采用c++语言进行程序编写。

1.3.4 c++界面设计：采用Qt实现程序界面展示。

2 需求分析

2.1 课题调研

最优化方法，是指解决最优化问题的方法。所谓最优化问题，指在某些约束条件下，决定某些可选择的变量应该取何值，使所选定的目标函数达到最优的问题。即运用最新科技手段和处理方法，使系统达到总体最优，从而为系统提出设计、施工、管理、运行的最优方案

本课题以人和任务作为元素，以资金作为权重，求解最优方案组合。

2.2 用户需求分析

程序需要满足如下需求：

1. 用户通过键盘输入规范数据，可以得到正确的计算结果，为用户提供最佳的方案建议。
2. 用户可以直接从文件里面获取数据。
3. 具有友好的交互界面，用户能简单直观的明白程序如何使用。
4. 结果输出界面，能清楚的展示最优的方案中的基本数据以及图形展示。
5. 对错误数据数据具有较好的识别能力，并向用户提供适当的反馈。
6. 程序具有调试功能。

3 方案设计

3.1 总体功能设计

程序的功能主要包括：输入功能，计算最佳任务分配方案和输出功能。

其中，输入功能包括文件输入及键盘输入两种输入方式。最佳任务分配方案采用匈牙利算法计算存在性、KM算法求解，得到资金最少的最优方案。输出功能是将经过核心算法处理后得到的结果输出到显示界面上，将会在显示框中显示所需的总资金、每个任务的方案选择。

程序主要功能图如图3.1所示：

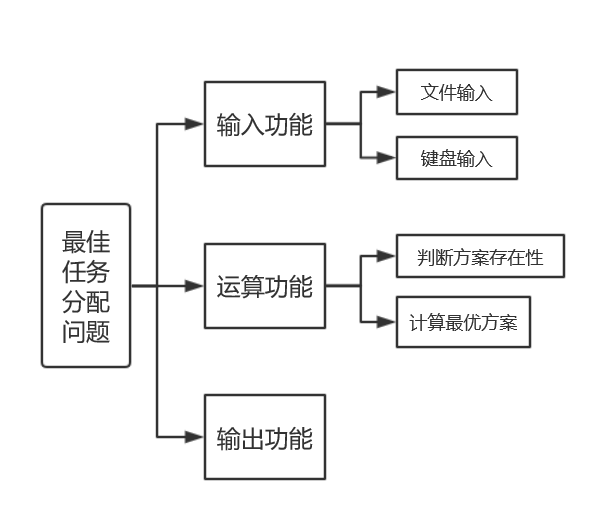


图 3.1 最佳任务分配功能图

3.2 数据结构设计

3.2.1 将数据结构及算法封装为KM类，包括存储权值的数据结构：二维向量矩阵。

|  |
| --- |
| KM |
| + matrix[105]:vector<double>  + matchx:vector<int>  + matchy:vector<int>  + mincost:double  + tn:int  + pn:int  + pname:vector<string>  + tname:vector<string>  - numx[105]:double  - numy[105]:double  - slack[105]:double  - visx[105]:int  - visy[105]:int  - d:double |
| + Find()  + \_km()  + dfs(int x)  + xyl()  + inputinit() |

3.3 函数原型设计

|  |  |
| --- | --- |
| 函数原型 | 功能描述 |
| bool inputinit() | 初始化 |
| bool txtinput(string &s) | 从键盘读入数据 |
| bool finput(string &path) | 从文件读入数据 |
| bool xyl() | 判断是否存在最佳方案 |
| bool Find(int v) | 匈牙利算法的子算法 |
| double \_km() | KM算法求解最优方案 |
| bool dfs(int x) | KM子算法 |

表3.3 函数原型设计表

3.4 主算法设计

3.4.1 匈牙利算法：

该算法目的是判断是否存在最优解。首先依次搜索任务寻找有人完成的可行路径，如果有边且未被访问，则记录下来，如果暂无匹配，或者原来匹配的左侧元素可以找到新的匹配，将当前做元素成为右侧元素的新匹配，如此循环直到无法找到更多的匹配，此时为二分图的最大匹配。

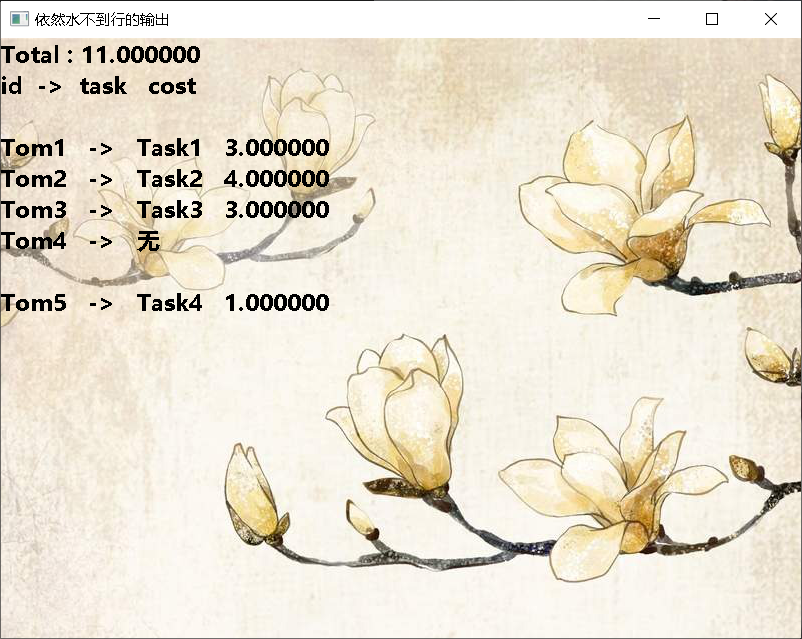
3.4.2 KM算法：

该算法用于寻找最优匹配方案。首先为每个结点初始化标杆，并依次搜索左元素为其寻找匹配，如果有边且未被访问，则记录下来，如果暂无匹配，或者原来匹配的左侧元素可以找到新的匹配，将当前做元素成为右侧元素的新匹配，如果找不到完备匹配，则修改标杆，并重复直到搜索到完备匹配时结束。

3.5 用户界面设计



输入页面



输出页面

3.6 输入输出设计

程序设计了文件输入和键盘输入两种输入方式，用户可根据需求选择对应的输入方式。根据输入的数据，经计算后输出最有任务分配方案。

4 方案实现

4.1 开发环境与工具

开发环境：Window 10

开发工具：CodeBlocks、Qt 5.4.2

4.2 程序设计关键技术

（1）KM算法

通过KM算法求出二分图的最优匹配及最优任务分配方案。

1. Qt可视化编程

Qt是一个开发的跨平台[C++](https://baike.baidu.com/item/C++" \t "https://baike.baidu.com/item/qt/_blank)[图形用户界面](https://baike.baidu.com/item/%E5%9B%BE%E5%BD%A2%E7%94%A8%E6%88%B7%E7%95%8C%E9%9D%A2" \t "https://baike.baidu.com/item/qt/_blank)应用程序开发框架。Qt是面向对象的框架，使用特殊的代码生成扩展（称为元对象编译器(Meta Object Compiler, moc)）以及一些宏，Qt很容易扩展，并且允许真正地组件编程。

4.3 个人设计实现（按组员分工）

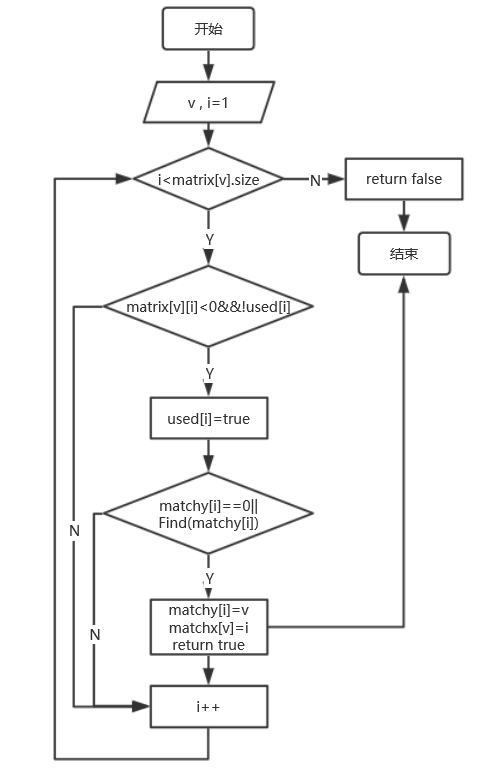
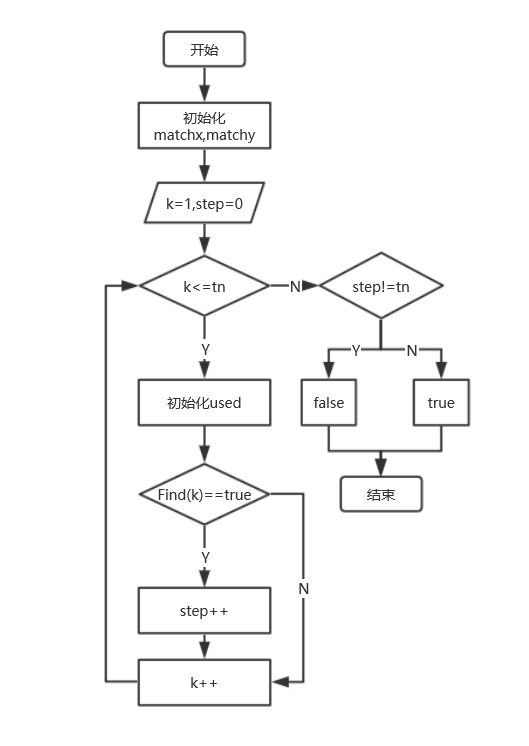
4.3.1 邱晓鹏设计实现

4.3.2 张琳设计实现

4.3.3 寇凯淇设计实现

（1）匈牙利算法：

流程图：



xyl() Find(int v)

代码：

bool KM::Find(int v){

for (int i = 1; i < matrix[v].size(); i++){

if (matrix[v][i] < 0 && !used[i]){

used[i] = true;

if (matchy[i] == 0 || Find(matchy[i])){

matchy[i] = v;

matchx[v] = i;

return true;

}

}

}

return false;

}

bool KM::xyl(){//分配

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

matchx.push\_back(0);

}

for (int n = 0; n <= pn; n++){

matchy.push\_back(0);

}

int step = 0;

for (int k = 1; k <= tn; k++){

memset(used, false, sizeof(used));

if (Find(k) == true){

step++;

//cout << step;

}

}

if (step != tn)

return false;

else

return true;

}

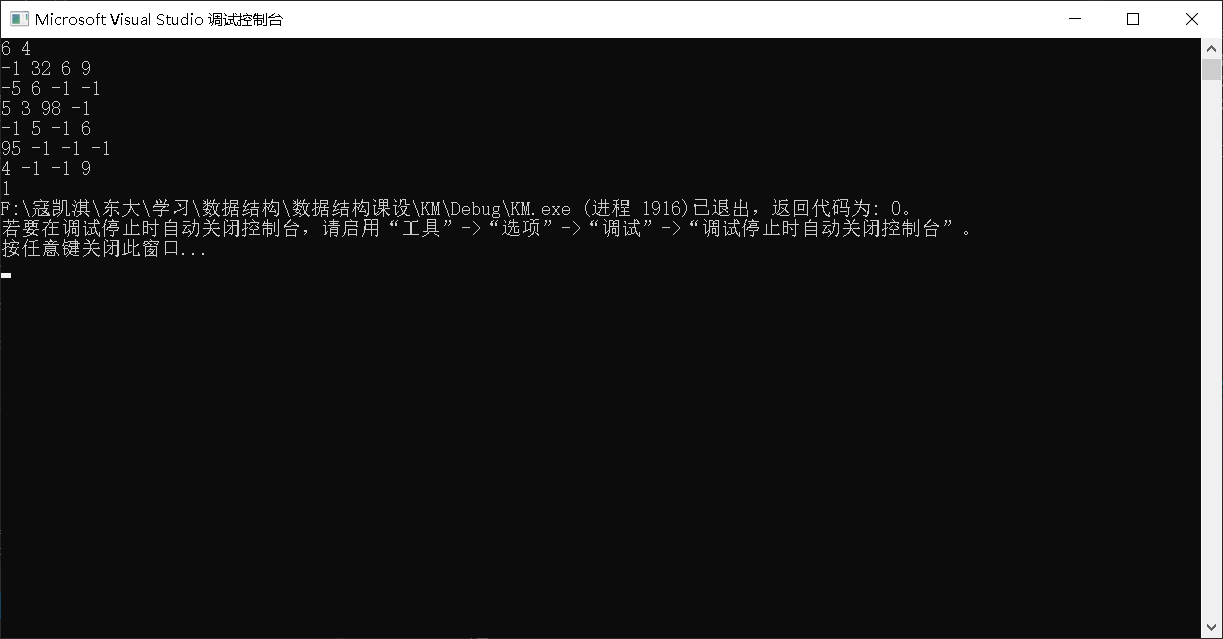
5 测试与调试

5.1 个人测试（按组员分工）

5.1.1 邱晓鹏测试

5.1.2 张琳测试

5.1.3 寇凯淇测试



5.2 组装与系统测试

将各个功能进行组装，程序能够稳定地运行，系统测试记录如下表所示。

|  |  |  |
| --- | --- | --- |
| 操作名称 | 操作流程 | 操作结果和输出 |
| 文件输入 | 点击文件输入按钮，选择存有数据信息的文件 | 成功从文件中读出数据，在输出界面成功输出数据，若数据有误或误解则输出没有解 |
| 键盘输入 | 在文本输入框中依次输入人数、任务数、依次的资金花费、人名及任务名 | 成功读入文本框中的信息，在输出界面成功输出数据和图示，若数据有误或误解则输出没有解 |

5.3 系统运行

系统能稳定运行并根据输入数据输出正确的结果。对于非法输入也能做出对应的反馈，提醒用户输入的数据不在合法范围内。界面友好，为用户带来较好的体验效果。



图 5.3.1 最优任务分配主界面

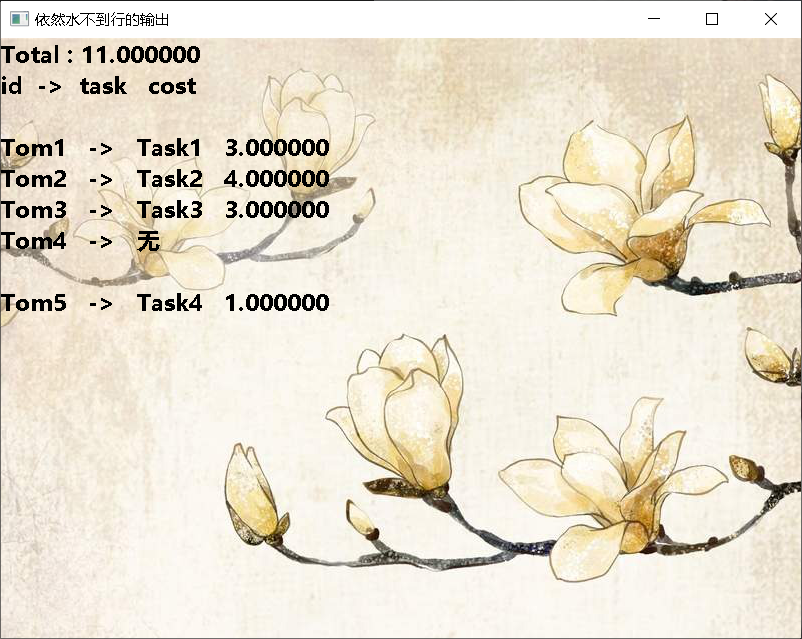


图5.3.2 输出界面

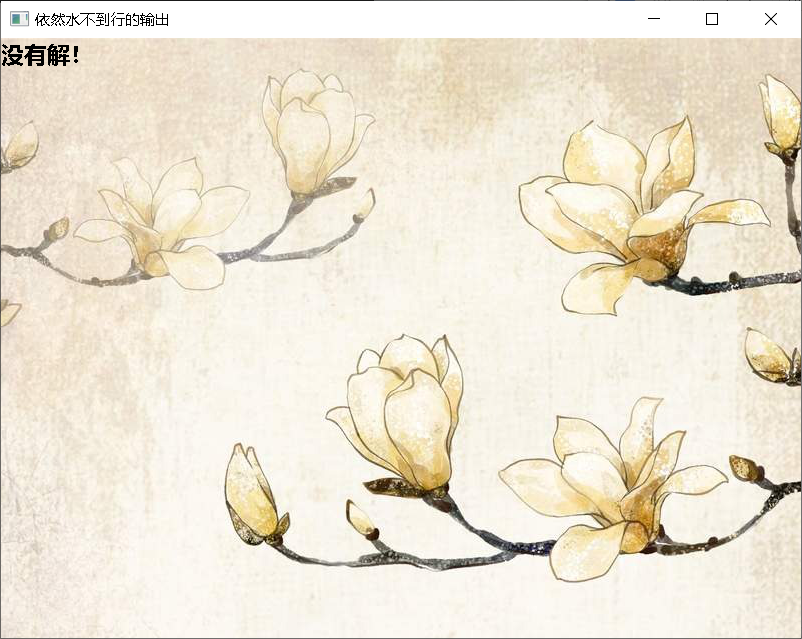


图5.3.3 非法输入输出界面

6 课题总结

6.1 课题评价

本课题用于解决最优分配方案问题，常运用最新科技手段和处理方法，使系统达到总体最优，从而为系统提出设计、施工、管理、运行的最优方案，具有很强的现实意义。通过输入具体的人、任务及对应的资金使用，可以计算出完成全部任务的最优匹配方案。

该最优任务分配系统实现了求解最优匹配的基本功能，具有计算求解迅速，能反复使用的优点。该系统在保证结果正确的基础上，还具有良好的稳定性，不会出现系统崩溃等异常情况，对非法输入也能进行合适的处理，最后，该程序具有良好的人机交互界面，使用便捷，输出结果清楚明了。

该课题也强化了团队成员队最优问题算法的认知与强化，丰富了理论知识与算法储备，提供了大量的实操经验 。

6.2 团队协作

课程设计的开始两节课，由组长组织组员进行讨论，对问题进行分析，决定程序中定义的数据结构、程序的基本功能、程序采用的算法。在程序数据结构确定下来以后，进行初步的分工，将程序的输入、算法实现、输出功能和界面设计分别分配给各个成员。各个成员间进行接口交流，保证各个部分的能很好的整合在一起，为整个程序服务。各成员在实现具体功能过程中也进行必要的沟通交流，并互相获取建议，在具体功能实现上遇到难题，共同寻找解决办法，遇到调试问题，大家也共同寻找原因。在实现自己负责部分的功能后，统一将程序各个部分整合在一起，并进行程序的测试和调试。

通过组内人员的共同努力，我们顺利完成了此次课程设计任务。通过团队合作，我们也更加熟悉团队合作开发的模式，注重团队间的交流，相互学习，共同进步。

6.3 个人设计小结（按组员分工）

6.3.1 邱晓鹏设计小结

6.3.2 张琳设计小结

6.3.3 寇凯淇设计小结

本次课设小组成员首先确定了各自的分工，并统一了各自的接口设计，使得本项目在编写合并调试的过程中，避免了很多不必要的麻烦，提高了代码编写测试的效率。

由于有第一个项目上在QT上的使用经验，此次的QT界面设计上有了很高的效率，完成的很迅速。此次项目的设计不但加深了我们对数据结构课堂上所授的结构与算法上的知识的认知，更丰富了我实践操作的经验，更通过实践扩展了很多课外的知识，使我受益匪浅。受限于时间与自身知识储备问题，此次的程序仍有很多未能优化的方面，我们也将会在之后继续对其进行优化，积累更多的知识与经验。

7 附录

A 课题任务分工

A-1 课题程序设计分工

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 程序设计函数原型、类 | 功能说明 |
|  | 邱晓鹏 |  |  |
|  | 张琳 |  |  |
| 20184446 | 寇凯淇 | bool xyl();  bool Find(int v); | (1)匈牙利算法 |

A-2 课题报告分工

|  |  |  |
| --- | --- | --- |
| 章节 | 内容 | 完成人 |
| 1 课题概述 | 1.1 课题任务  1.2 课题原理  1.3 相关知识 |  |
| 2 需求分析 | 2.1 课题调研  2.2 用户需求分析 |  |
| 3 方案设计 | 3.1 总体功能设计  3.2 数据结构设计  3.3 函数原型设计  3.4 输入输出设计  3.5 主算法设计  3.6 用户界面设计 |  |
| 4 方案实现 | 4.1 开发环境与工具  4.2 程序设计关键技术  4.3 个人设计实现（按组员分工）  4.3.1  4.3.2  4.3.3 |  |
| 5 测试与调试 | 5.1 个人测试（按组员分工）  5.1.1  5.1.2  5.1.3  5.2 组装与系统测试  5.3 系统运行 |  |
| 6 课题总结 | 6.1 课题评价  6.2 团队协作  6.3 下一步工作  6.4 个人设计心得（按组员分工）  6.4.1  6.4.2  6.4.3 |  |

B 课题设计文档（光盘）

B-1课程设计报告（电子版）

B-2源程序代码（\*.H，\*.CPP）

//input.h

#ifndef INPUT

#define INPUT

#include<iostream>

#include<algorithm>

#include<sstream>

#include<fstream>

#include<string>

#include<QDebug>

using namespace std;

bool txtinput(string &s);

bool finput(string &path);

#endif // INPUT

//km.h

#ifndef KM\_H

#define KM\_H

#include<iostream>

#include<algorithm>

#include<vector>

#include<string>

#include<cstring>

using namespace std;

class KM

{

public:

KM();

vector<double> matrix[105];

vector<int> matchx;

vector<int> matchy;

double mincost;

int tn,pn;

bool used[105];

vector<string> pname;

vector<string> tname;

bool Find(int v);

double \_km();

bool dfs(int x);

bool xyl();

bool inputinit();

private:

double numx[105],numy[105];

double slack[105];

int visx[105],visy[105];

double d;

};

extern KM ans;

extern bool ifok;

#endif // KM\_H

//mainwindow.h

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QPushButton>

#include <QTextEdit>

#include <QString>

#include <QDebug>

#include <QFileDialog>

#include <QPixmap>

#include <QPalette>

#include "output.h"

namespace Ui {

class MainWindow;

}

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

explicit MainWindow(QWidget \*parent = 0);

~*MainWindow*();

void txtin();

void fcin();

QString openfile();

void ofput();

private:

Ui::MainWindow \*ui;

output \*oput=NULL;

QPalette palette;

};

#endif // MAINWINDOW\_H

//output.h

#ifndef OUTPUT\_H

#define OUTPUT\_H

#include <QWidget>

#include <QLabel>

#include <QDebug>

namespace Ui {

class output;

}

class output : public QWidget

{

Q\_OBJECT

public:

explicit output(QWidget \*parent = 0);

~*output*();

private:

Ui::output \*ui;

QLabel \*qout;

QPalette back1;

};

void temp();

#endif // OUTPUT\_H

//input.cpp

#include "input.h"

#include "km.h"

bool txtinput(string &s){

ans.inputinit();

double temp;

stringstream ss(s);

if(!(ss>>ans.pn>>ans.tn)){

ans.pn=ans.tn=0;

return false;

}

for(int i=1;i<=ans.pn;i++){

for(int j=1;j<=ans.tn;j++){

if(i==1){

(ans.matrix[j]).push\_back(1);

}

if(!(ss>>temp)){

return false;

}

(ans.matrix[j]).push\_back((-1)\*temp);

}

}

ans.pname.push\_back("\0");

ans.tname.push\_back("\0");

string name;

for(int i=0;i<ans.pn;i++){

if(!(ss>>name)){

return false;

}

ans.pname.push\_back(name);

}

for(int i=0;i<ans.tn;i++){

if(!(ss>>name)){

return false;

}

ans.tname.push\_back(name);

}

return true;

}

bool finput(string &path){

ans.inputinit();

ifstream ss(path);

double temp;

if(!(ss>>ans.pn>>ans.tn)){

ans.pn=ans.tn=0;

return false;

}

for(int i=1;i<=ans.pn;i++){

for(int j=1;j<=ans.tn;j++){

if(i==1){

(ans.matrix[j]).push\_back(1);

}

if(!(ss>>temp)){

return false;

}

(ans.matrix[j]).push\_back((-1)\*temp);

}

}

ans.pname.push\_back("\0");

ans.tname.push\_back("\0");

string name;

for(int i=0;i<ans.pn;i++){

if(!(ss>>name)){

return false;

}

ans.pname.push\_back(name);

}

for(int i=0;i<ans.tn;i++){

if(!(ss>>name)){

return false;

}

ans.tname.push\_back(name);

}

return true;

}

//km.cpp

#include "km.h"

KM::KM()

{

}

KM ans;

bool ifok=true;

const int inf=2e9+7;

bool KM::Find(int v)

{

/\*used[v] = true;

for (int i = 1; i <= matrix[v].size(); i++)

{

int u = matrix[v][i];

int w = matchx[u];

if (w < 0 || !used[w] && Find(w))

{

matchx[v] = u;

matchy[u] = v;

return true;

}

}

return false;\*/

for (int i = 1; i < matrix[v].size(); i++)

{

if (matrix[v][i] < 0 && !used[i])

{

used[i] = true;

if (matchy[i] == 0 || Find(matchy[i]))

{

matchy[i] = v;

matchx[v] = i;

return true;

}

}

}

return false;

}

bool KM::xyl()//分配

{

for (int m = 0; m <= tn; m++)

{

matchx.push\_back(0);

}

for (int n = 0; n <= pn; n++)

{

matchy.push\_back(0);

}

int step = 0;

for (int k = 1; k <= tn; k++)

{

memset(used, false, sizeof(used));

if (Find(k) == true)

{

step++;

//cout << step;

}

}

if (step != tn)

return false;

else

return true;

}

bool KM::dfs(int x)

{

visx[x]=1;

for(int i=1; i<=pn; i++)

{

if(numx[x]+numy[i] == matrix[x][i])

{

if(!visy[i])

{

visy[i]=1;

if(!matchy[i] || dfs(matchy[i]))

{

matchy[i]=x;

matchx[x]=i;

return 1;

}

}

}

else //计算当前的最小减少点标

slack[i]=min(slack[i],numx[x]+numy[i]-matrix[x][i]);

}

return 0;

}

double KM::\_km()

{

for(int i=1; i<=tn; i++) //点标初始化

{

numx[i] = -inf;

}

memset(numy,0,sizeof(numy));

for(int i=0; i<=tn; i++)

matchx[i]=0;

for(int i=0; i<=pn; i++)

matchy[i]=0;

for(int i=1; i<=tn; i++) //给每一个点的初始化点标

for(int j=1; j<=pn; j++)

if(matrix[i][j]<0)

numx[i] = max(numx[i],matrix[i][j]);

else matrix[i][j]=-inf;

for(int i=1; i<=tn; i++)

{

while(1)

{

for(int i=1; i<=105; i++)

{

slack[i] = inf;

}

memset(visx,0,sizeof(visx));

memset(visy,0,sizeof(visy));

if(dfs(i))

break;

d = inf;

for(int j=1; j<=pn; j++)

{

if(!visy[j])

{

d=min(d,slack[j]);

}

}

for(int j=1; j<=max(tn,pn); j++)

{

if(visy[j])

numy[j]+=d;

if(visx[j])

numx[j]-=d;

}

}

}

mincost = 0;

for(int i=1; i<=tn; i++)

{

mincost += matrix[i][matchx[i]];

}

mincost = -mincost;

for(int i=1; i<=tn; i++)

cout<<"第"<<matchx[i]<<"个人配第"<<i<<"项课题"<<endl;

cout<<"一共需要"<<mincost<<"yuan"<<endl;

return mincost;

}

bool KM::inputinit(){

for(int i=0;i<105;i++){

matrix[i].clear();

}

matchx.clear();

matchy.clear();

tn=pn=0;

pname.clear();

tname.clear();

}

//main.cpp

#include "mainwindow.h"

#include <QApplication>

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

MainWindow w;

w.setWindowTitle("最佳任务分配");

w.show();

return a.exec();

}

//mainwindow.cpp

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include "km.h"

#include "input.h"

MainWindow::MainWindow(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow)

{

this->autoFillBackground();

palette.setBrush(QPalette::Background,QBrush(QPixmap("shu.jpg")));

this->setPalette(palette);

ui->setupUi(this);

ui->txt->setPlaceholderText("这里是输入框\n"

"请输入人数、任务的数目：\n"

"然后依次输入每一个人对每一项任务完成的花费\n"

"-1为不可完成\n"

"最后输入每一个人的名字、以及每一项任务的名称");

connect(ui->in,QPushButton::clicked,this,&txtin);

connect(ui->fcin,QPushButton::clicked,this,&fcin);

connect(ui->pushButton,QPushButton::clicked,this,&std::exit);

}

MainWindow::~*MainWindow*()

{

delete ui;

}

void MainWindow::txtin(){

QString qtxt = ui->txt->toPlainText();

string txt;

txt=qtxt.toStdString();

cout<<txt<<endl;

ifok=txtinput(txt);

if(ifok==true)

{

ifok=ans.xyl();

ans.\_km();

}

ofput();

}

void MainWindow::fcin(){

string path=openfile().toStdString();

ifok=finput(path);

if(ifok==true)

{

ifok=ans.xyl();

ans.\_km();

}

ofput();

}

QString MainWindow::openfile(){

QString path;

//qDebug() << "?";

path=QFileDialog::getOpenFileName(this,"open","./","\*.txt");

//path = QDir::toNativeSeparators(path);

int y = 0;

y = path.indexOf(QString("/"));

//qDebug()<<y<<"\\";

while(y!=-1)

{

path.replace(y,1,QString("\\\\"));

y = path.indexOf(QString("/"));

qDebug()<<y;

}

qDebug() << path;

return path;

}

void MainWindow::ofput(){

oput=new output;

oput->setWindowTitle("依然水不到行的输出");

oput->show();

}

//output.cpp

#include "output.h"

#include "ui\_output.h"

#include "km.h"

output::output(QWidget \*parent) :

QWidget(parent),

ui(new Ui::output)

{

ui->setupUi(this);

//temp();

this->autoFillBackground();

back1.setBrush(QPalette::Background,QBrush(QPixmap("shu.jpg")));

this->setPalette(back1);

QPalette pa;

pa.setColor(QPalette::WindowText,Qt::black);

QFont ft("Microsoft YaHei",14,75);

//ft.setPointSize(14);

qout=new QLabel;

qout->setFont(ft);

qout->setPalette(pa);

QString qs;

qout->setParent(this);

if(!ifok){

qs+="没有解！\n";

}

else {

qs="Total : ";

qs+=QString::fromStdString(std::to\_string(ans.mincost));

qs+="\nid -> task cost\n";

qDebug()<<ans.pn;

for(int i=1;i<=ans.pn;i++){

qs+='\n';

qs+=QString::fromStdString(ans.pname[i]);

qs+=" -> ";

if(ans.matchy[i]==0){

qs+="无\n";

continue;

}

qs+=QString::fromStdString(ans.tname[ans.matchy[i]]);

qs+=" ";

qs+=QString::fromStdString(std::to\_string((-ans.matrix[ans.matchy[i]][i])));

}

}

qDebug()<<qs;

qout->setText(qs.toUtf8());

qout->show();

this->show();

}

output::~*output*()

{

delete ui;

delete qout;

}

void temp(){

for(int i=0;i<ans.pn+1;i++){

ans.matchy.push\_back(i);

}

for(int i=0;i<ans.tn+1;i++){

ans.matchx.push\_back(i);

}

ans.mincost=3.5;

}

B-3工程与可执行文件

C.1 运行环境说明

程序推荐在Windows环境下使用。

C.2 操作说明

进入程序主界面，根绝个人使用需求，选择键盘输入到文本框或文件输入。对于键盘输入，需要按照提示格式输入，否则会被非法输入处理，输入结束后点击输入按钮获取结果。对于文件输入，点击文件按钮选取合法数据的文件获得相应结果。查询结束后，关闭输出结果界面。程序使用完毕，点击退出按钮关闭主菜单界面、退出程序。