**云南大学数学系《离散数学》上机实验报告**

|  |  |  |
| --- | --- | --- |
| **课程名称**：离散数学 | **学期：**2016-2017学年秋季学期 | **成绩**： |
| **指导教师**：李建平 | **学生姓名**：刘鹏 | **学生学号**：20151910042 |
| **实验名称：**[7]Spanning\_Tree\_of\_a\_relation | | |
| **实验编号**：No.7 | **实验日期**：2016年10月27日 | **实验学时**：2 |
| **学院：**数学与统计学院 | **专业：**信息与计算科学 | **年级**：2015级 |

# 一、实验目的

用C语言进行算法实现。

# 二、实验内容

# 三、使用环境

**编译环境：**

Windows10 Enterprise中文版操作系统，

Code::Blocks 16.01编译器。

**使用语言：C**

# 四、算法介绍

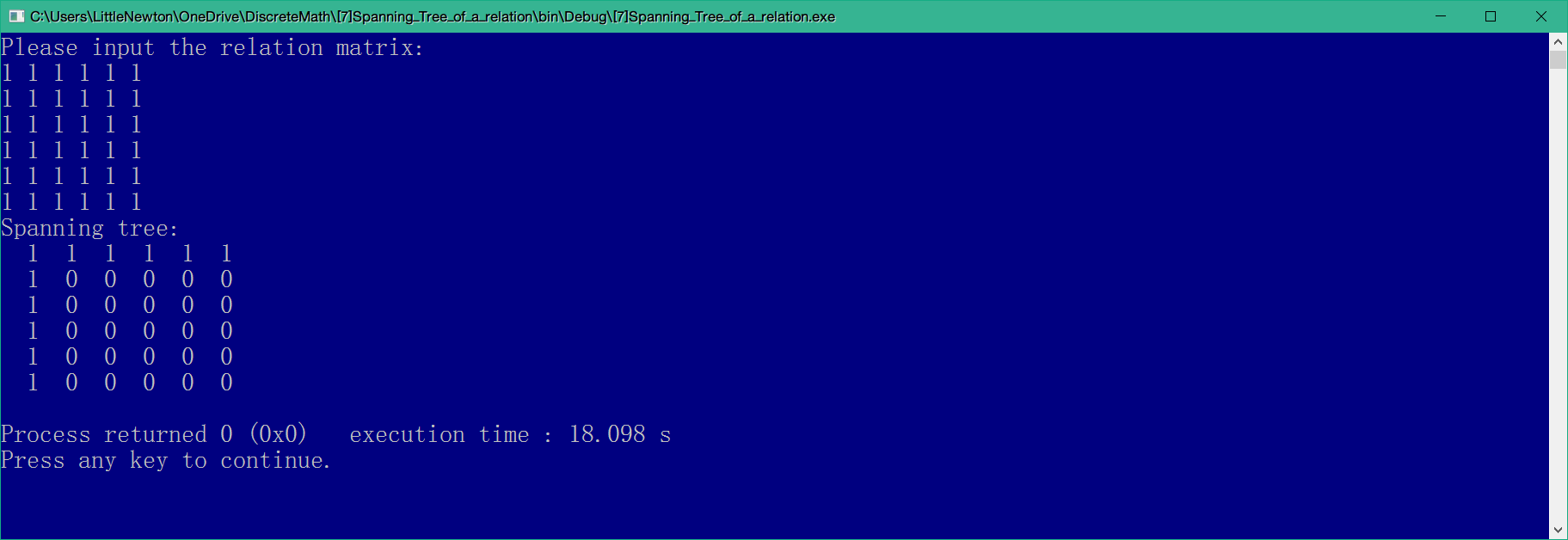
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | **Algorithm** Prim  **Input**: The matrix  of relation R defined on set  **Output**: Whether R a connected relation or not, if true, give the spanning tree of R by matrix  **Begin**  **Step 1**: calculate the adjacency matrix = of  for i =0 throgh n  for j=0through n  if( and )then    for i=0 through n  for j=0 through n  for k=0 through n  if( or( and )) then    **Step 2**: for i=0 through n  for j=0 through n  if  then  output: R is not connected  STOP  **Step 3**: choose  and ,  Choose another vertex  in which is connected with  Let , and  **Step 4**: repeat step 3 until , then get the spanning tree  **End** |

# 五、调试过程

程序代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53 | /\*  \* Copyright 2016, LittleNewton  \* All rights reserved  \*  \* filename: [7]Spanning\_Tree\_of\_a\_relation  \*/  #include <stdio.h>  #define N 6  int main**()**  **{**  int relation\_matrix**[**N**][**N**];**  int spanning\_tree**[**N**][**N**]={{**0**}};**  int guard**[**N**]={**0**};**  int i**,**j**;**  printf**(**"Please input the relation matrix:\n"**);**  **for(**i**=**0**;**i**<**N**;**i**++)**  **{**  **for(**j**=**0**;**j**<**N**;**j**++)**  **{**  scanf**(**"%d"**,&**relation\_matrix**[**i**][**j**]);**  **}**  **}**  **for(**i**=**0**;**i**<**N**;**i**++)**  **{**  **for(**j**=**0**;**j**<**N**;**j**++)**  **{**  **if(**relation\_matrix**[**i**][**j**]==**1 **&&** guard**[**j**]==**0**)**  **{**  spanning\_tree**[**i**][**j**]=**1**;**  spanning\_tree**[**j**][**i**]=**1**;**  guard**[**j**]=**1**;**  **}**  **}**  **}**  **for(**i**=**0**;**i**<**N**;**i**++)**  **{**  **if(**guard**[**i**]==**0**)**  **{**  printf**(**"This directed graph is not connected."**);**  **return** 0**;**  **}**  **}**  printf**(**"Spanning tree:\n"**);**  **for(**i**=**0**;**i**<**N**;**i**++)**  **{**  **for(**j**=**0**;**j**<**N**;**j**++)**  **{**  printf**(**"%3d"**,**spanning\_tree**[**i**][**j**]);**  **}**  printf**(**"\n"**);**  **}**  **return** 0**;**  **}** |

2.运行结果



# 六、总结

算法与代码相关性不大，导致编程比较困难。从局部连通性成立的假设以及树的结构中考虑，可以比较快速写出算法并且判断连通性。

# 七、参考文献

[1] 谭浩强，C程序设计（第四版），清华大学出版社，清华大学，2015年6月

[2] Bernard Kolman, Robert C. Busby and Sharon Cutler Ross, *Discrete Mathematical Structures*, Pearson Education, Inc

# 八、教师评语