DDB2011Prototype

关键词查询数据库

开发文档

小组编号：第2组

小组成员：刘硕 罗云 王旭 王旭波

# 目录

[1 引言 4](#_Toc309471298)

[1.1 程序功能 4](#_Toc309471299)

[1.2 参考资料 4](#_Toc309471300)

[2 程序设计 4](#_Toc309471301)

[2.1 开发环境 4](#_Toc309471302)

[2.2 数据库结构 4](#_Toc309471303)

[2.3 算法设计 5](#_Toc309471304)

[2.4 数据结构设计 7](#_Toc309471305)

[2.5 界面设计 7](#_Toc309471306)

[3 程序实现 8](#_Toc309471307)

[3.1 程序逻辑部分接口及数据结构实现 8](#_Toc309471308)

[SteinerTree.cs 8](#_Toc309471309)

[Util.cs 9](#_Toc309471310)

[GraphManager.cs 9](#_Toc309471311)

[Dijkstra.cs 10](#_Toc309471312)

[DBManager.cs 10](#_Toc309471313)

[3.2 程序界面部分接口及数据结构实现 11](#_Toc309471314)

[DataGraphGeometry.cs 11](#_Toc309471315)

[IDataGraphGeometry.cs 13](#_Toc309471316)

[MainWindow.cs 14](#_Toc309471317)

# 1 引言

## 1.1 程序功能

采用基于数据图的方法，设计并实现一个“关系数据库中的关键词查询”原型系统，用户输入查询关键词，查询引擎根据查询关键词到数据库图中寻找一颗最小Steiner 树。

## 1.2 参考资料

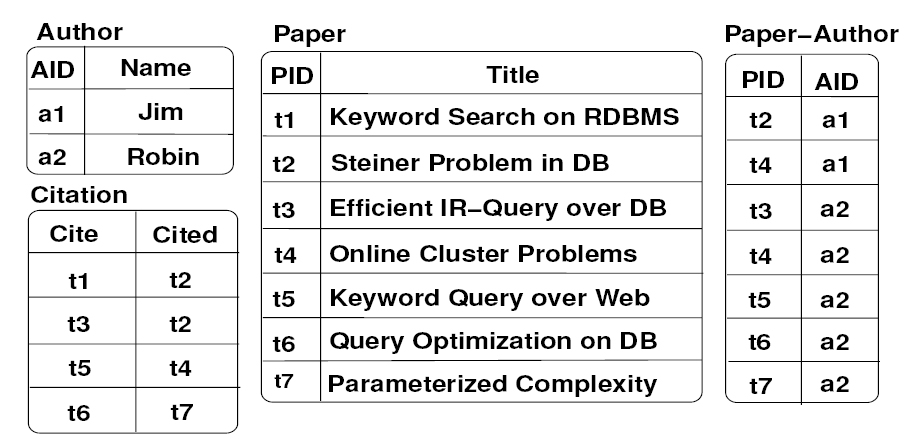
* **基于关键数据库的关键词查询**
* **Keyword Searching and Browsing in Databases using BANKS**
* [**http://blog.csdn.net/jdream314/article/details/6572178**](http://blog.csdn.net/jdream314/article/details/6572178)

# 2 程序设计

## 2.1 开发环境

* **Windows 7操作系统**
* **Microsoft .NET Framework 4**
* **Microsoft Visual Studio 2010**
* **MySQL数据库**

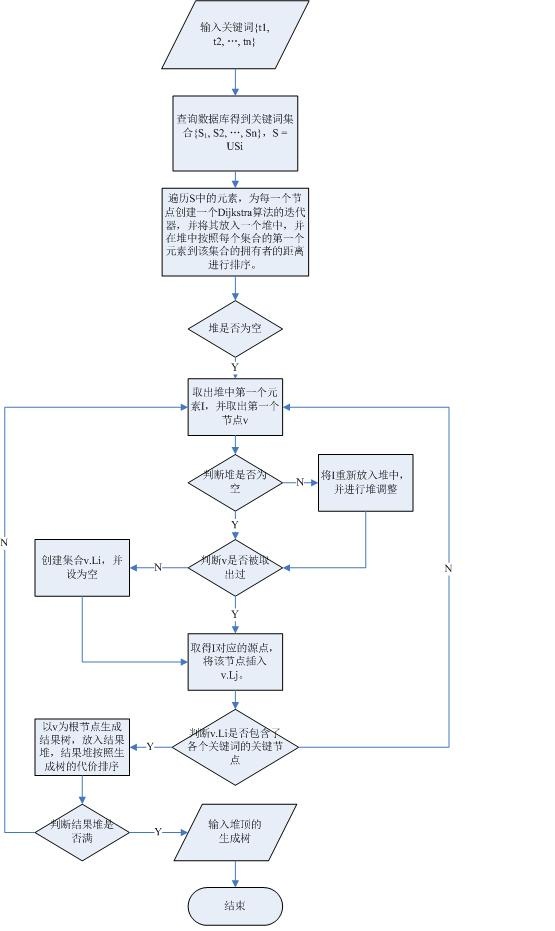
## 2.2 数据库结构



## 2.3 算法设计

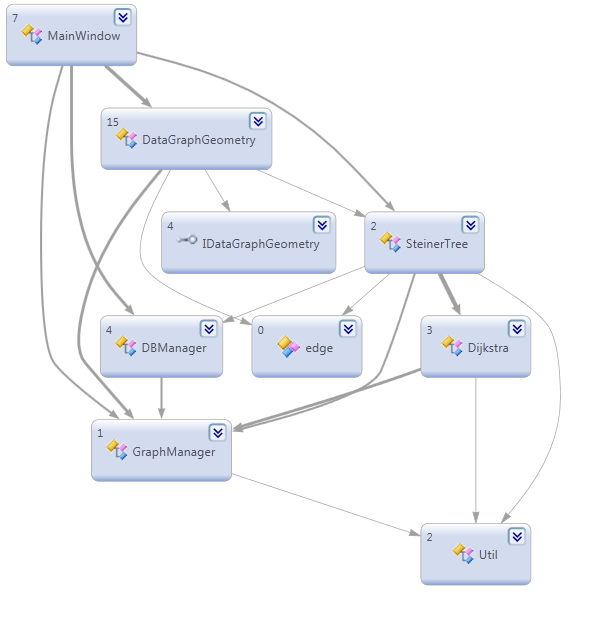
本程序涉及的算法主要是steiner 树的生成算法，我们主要利用了BANKS算法的思想，求解steiner树。

算法流程图如下：

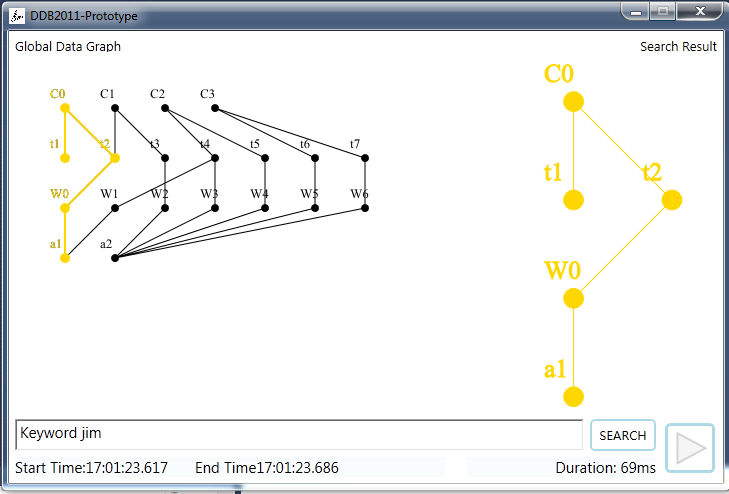


## 2.4 数据结构设计

设计了一系列的类，其主要依赖关系如下图：



## 2.5 界面设计



界面包括数据图显示区、Streiner树显示区、关键词搜索框、搜索按钮以及显示算法执行时间的状态条。

# 3 程序实现

## 3.1 程序逻辑部分接口及数据结构实现

### SteinerTree.cs

public class **SteinerTree**

public **SteinerTree**(**GraphManager** *gm***, DBManager** *dbm***, string[]** *keyword*)

**Summary:**

构造函数

**Parameters:**

*gm*: 图结构

*dbm*: 数据库管理类

*keyword*: 关键词集合

public **void** **execute**()

**Summary:**

算法执行

Public **List<int>** **steinerTreeNode**

**Summary:**

steiner树的节点集合

public **List<edge>** **steinerTreeEdge**

**Summary:**

steiner树的边集合

### Util.cs

public static readonly **int** **INFINITE**

**Summary:**

32位机的最大整数，用于表示无穷大

### GraphManager.cs

public **class** GraphManager

**Summary**:

此类用于保存图的各种数据结构以及算法

public **GraphManager**(**int** *nodeNum*)

**Summary**:

构造函数

**Parameters**:

*nodeNum*: 节点数目

public **System.Array** arrayKey

**Summary**:

根据节点的索引得到节点的主键

public **System.Array** arrayPos

**Summary**:

绘图时用于记录每一个点的位置

public **Dictionary<string,int>** dicKey

**Summary**:

根据节点的主键得到节点的索引

public **int[,]** graph

**Summary**:

此乃保存所有节点关系的矩阵，即图的矩阵

public **int** nodeNum

**Summary**:

点的个数

### Dijkstra.cs

internal class **Dijkstra**

**Summary:**

执行Dijkstra算法

public **Dijkstra**(**GraphManager** *gm***, int** *v0*)

**Summary:**

构造函数

**Parameters:**

*gm*:

public **void** **execute**()

**Summary:**

算法初始化执行

public **int** **getNextNode**()

**Summary:**

算法的迭代执行

**Returns:**

下一个节点编号

public **int** **nextDistance**

**Summary:**

迭代器得到下个节点到源节点的距离

public **int** **nextV**

**Summary:**

迭代器得到的下一个节点

public **int[]** **pre**

**Summary:**

记录最短路径中该节点的上一个节点号

public **int** **v0**

**Summary:**

源节点

### DBManager.cs

public class **DBManager**

**Summary:**

此类用于处理数据库的相关操作

public **List<int>** **getKeywordSets**(**string[]** *keyword***, GraphManager** *gm***, List<List<int>>** *keywordSet*)

**Summary:**

根据关键词获得节点集合

**Parameters:**

*keyword*:

*gm*:

public **int** **getNodeNum**()

**Summary:**

获得数据库中所有节点的数目

**Returns:**

节点数目

public **void** **initGraph**(**GraphManager** *gm*)

**Summary:**

初始化数据图

**Parameters:**

*gm*:

## 3.2 程序界面部分接口及数据结构实现

### DataGraphGeometry.cs

public class **DataGraphGeometry**

Member of **DDB2011Prototype**

**Summary:**

用于存储数据图几何图形的类，需要实现IDataGraphGeometry以实现和GraphManager类的对接，负责完成将GraphManager类向PathGeometry类的转化。

public **int** **graphID** { set; get; }

Member of **DDB2011Prototype**.**DataGraphGeometry**

**Summary:**

图的唯一编号

public **string** **graphName** { set; get; }

Member of **DDB2011Prototype**.**DataGraphGeometry**

**Summary:**

图的名字

public **System.Windows.Media.PathGeometry** **dataGraphLine**

Member of **DDB2011Prototype**.**DataGraphGeometry**

**Summary:**

最终返回用到的几何图形组

public **System.Windows.Media.PathGeometry** **dataGraphShape**

Member of **DDB2011Prototype**.**DataGraphGeometry**

**Summary:**

最终返回用到的几何图形组

public **System.Windows.Media.PathGeometry** **steinertreeLine**

Member of **DDB2011Prototype**.**DataGraphGeometry**

**Summary:**

最终返回用到的几何图形组

public **System.Windows.Media.PathGeometry** **steinertreeShape**

Member of **DDB2011Prototype**.**DataGraphGeometry**

**Summary:**

最终返回用到的几何图形组

public **System.Windows.Media.PathGeometry[]** **ConstructGeometrySimple**(**DDB2011Prototype.GraphManager** *gm***, int** *radius***, int** *fontsize*)

Member of **DDB2011Prototype**.**DataGraphGeometry**

**Summary:**

从数据图构造几何图形

**Parameters:**

*gm*: 数据图GraphManager

*radius*: 每個點的半徑

*fontsize*: 字体大小

public **System.Windows.Media.PathGeometry[]** **ConstructSteinterGeometry**(**DDB2011Prototype.SteinerTree** *st***, DDB2011Prototype.GraphManager** *gm***, int** *radius***, int** *fontsize*)

Member of **DDB2011Prototype**.**DataGraphGeometry**

**Summary:**

构造steiner树的几何图形

**Parameters:**

*st*: steinertree类

*gm*: GraphManager类

*radius*: 每个点的半径

*fontsize*: 每个点说明文字的大小

**Returns:**

几何图形组。[0]为点集，[1]为边集

private **System.Windows.Media.GlyphRun** **DataGraphPrintText**(**string** *s***, System.Windows.Point** *p***, int** *fontsize*)

Member of **DDB2011Prototype**.**DataGraphGeometry**

**Summary:**

在指定点画出文字

**Parameters:**

*s*: 所画的文字内容

*p*: 文字坐标

*fontsize*: 文字大小

**Returns:**

返回文字形状

### IDataGraphGeometry.cs

DataGraphGeometry需实现的接口，用于完成和GraphManager的对接

**System.Windows.Media.PathGeometry[]** **ConstructGeometrySimple**(**DDB2011Prototype.GraphManager** *gm***, int** *radius***, int** *fontsize*)

Member of **DDB2011Prototype**.**IDataGraphGeometry**

**Summary:**

DDB2011专用简化绘图

**Parameters:**

*gm*: GraphManager数据图

*radius*: 每個點的半徑

*fontsize*: 文字大小

**Returns:**

返回几何图形集

**System.Windows.Media.PathGeometry[]** **ConstructSteinterGeometry**(**DDB2011Prototype.SteinerTree** *st***, DDB2011Prototype.GraphManager** *gm***, int** *radius***, int** *fontsize*)

Member of **DDB2011Prototype**.**IDataGraphGeometry**

**Summary:**

SteinerTree类专用单独绘图

**Parameters:**

*st*: 最简查询子树

*gm*: GraphManager数据图

*radius*: 每個點的半徑

*fontsize*: 文字大小

**Returns:**

几何图形

### MainWindow.cs

public class **MainWindow** : **System.Windows.Window**

Member of **DDB2011Prototype**

**Summary:**

接管MainWindow.xaml的逻辑部分

internal **System.Windows.Controls.Grid** **canvas**

Member of **DDB2011Prototype**.**MainWindow**

**Summary**

窗口上的主绘图区

internal **System.Windows.Controls.Grid** **canvasTree**

Member of **DDB2011Prototype**.**MainWindow**

**Summary**

窗口右侧的查询树绘图区

internal **System.Windows.Controls.Button** **buttonStartDraw**

Member of **DDB2011Prototype**.**MainWindow**

**Summary**

绘制数据图按钮

internal **System.Windows.Controls.Button** **buttonGO**

Member of **DDB2011Prototype**.**MainWindow**

**Summary**

查询按钮

internal **System.Windows.Controls.TextBlock** **textBlockInfo**

Member of **DDB2011Prototype**.**MainWindow**

**Summary**

显示时间文本

internal **System.Windows.Controls.TextBlock** **textBlockInfo2**

Member of **DDB2011Prototype**.**MainWindow**

**Summary**

显示时间文本

internal **System.Windows.Controls.TextBox** **textBoxSearch**

Member of **DDB2011Prototype**.**MainWindow**

**Summary**

用于输入查询关键词

private **void** **buttonStartDraw\_Click**(**object** *sender***, System.Windows.RoutedEventArgs** *e*)

Member of **DDB2011Prototype**.**MainWindow**

**Summary:**

单击显示数据图按钮的相应事件

**Parameters:**

*sender*:

*e*:

private **void** **buttonGO\_Click**(**object** *sender***, System.Windows.RoutedEventArgs** *e*)

Member of **DDB2011Prototype**.**MainWindow**

**Summary:**

单击搜索按钮的响应

**Parameters:**

*sender*:

*e*:

private **void** **imageDisplay**(**System.Windows.Media.PathGeometry[]** *dataGraph***, System.Windows.Media.Brush** *brush***, System.Windows.Controls.Grid** *grid***, bool** *uniform***, int** *lineThickness***, bool** *shapeStroke*)

Member of **DDB2011Prototype**.**MainWindow**

**Summary:**

将数据图绘制到屏幕

**Parameters:**

*dataGraph*: GraphManager数据图

*brush*: 笔刷 用于调节颜色

*grid*: Grid 用于指定绘图板

*uniform*: 用于指定是否自动缩放

*lineThickness*: 用于指定线条宽度

*shapeStroke*: 用于指定是否绘制边线