Code Project 4—Johnson 最短路径对

1. **原理概述**
   1. Bellman-Ford 算法

Bellman-Ford算法用于求解单源最短路径。其思想是，通过对图中每一条边的|G.V|-1轮松弛操作，每个节点的d值收敛到其到达源点的最短路径值。在每一步松弛操作下，更新节点的前驱节点。

Bellman-Ford算法允许边的权值为负值，但是要求图中不存在权值为负的环路。

* 1. Dijkstra 算法

Dijkstra算法也被用于求解单源最短路径。要求图中没有权值为负的边。Dijkstra算法通过维护一个最小堆来实现每次对一个节点的出边进行松弛操作。算法首先初始化一个记录节点到达源节点的权值的列表，把除源节点以外所有节点的d值设置为∞，源节点设置为0。在循环体中每次从最小堆（key为每个节点的d值）取出key最小的节点，对节点的邻接节点进行松弛操作，并把这个节点加入到另一个已完成操作的集合中。直到最小堆为空，算法结束。

* 1. 对稀疏图的Johnson算法——所有节点之间的最短路径

Johnson算法首先对边的权值有负值的图进行权重重赋值操作。这个操作通过向图中加入一个新的节点，这个节点生成指向之前所有节点的边，权值为0。之后对这个新节点运行bellman-Ford算法，用算法的结果给每个节点赋值h(u)=δ(u,s)。之后对边重赋值，使得边的结果没有负值。这样我们就可以对每个节点运行Dijkstra算法算的所有节点对之间的最短路径。

1. **代码实现：**

我们新建VS2010工程，新建5个类；

* 1. graph：

这个类描述了基本的图。采用邻接链表表示。相关的类方法例如：InitEdge()这个方法随机初始化一个图。调用内部的InsertEdge等方法实现。对于邻接链表的每个节点，我们新定义了一个结构体，内部包含目标节点的index，权值以及下一个节点的指针。

* 1. SingleSrcShortest:

这个类是单源最短路径的基类。定义了一个结构体djn，结构体包含当前节点的index，节点的d值以及节点的前驱节点的index。这个基类用一个djn数组作为成员变量。另外定义了类方法，一个是InitSingleSrc(int src) 这个方法用于初始化，把所有节点的前驱节点index设置为-1，表示没有；把所有节点的d值设置为BIGNUMBER，用于模拟正无穷。BIGNUMBER是一个宏定义。另一个类方法是Relax，用于松弛操作。还定义了打印方法用于输出最短路径树。

* 1. Bellman-Ford：

这个类用于运行BF算法。继承自父类SingleSrcShortest。

* 1. Dijkstra

这个类运行Dijkstra算法。继承父类SingleSrcShortest。这里简化了最小堆操作，采用暴力搜索算法。

* 1. Johnson

运行Johnson算法。用一个djn二维矩阵来存储pi矩阵和d矩阵。另设一个djn数组用于前期跑bellman-ford算法以及权值重新赋值操作。

* 1. 主程序：先随机生成图，测试bf算法，之后跑Johnson算法。并且打印
  2. 项目类图：



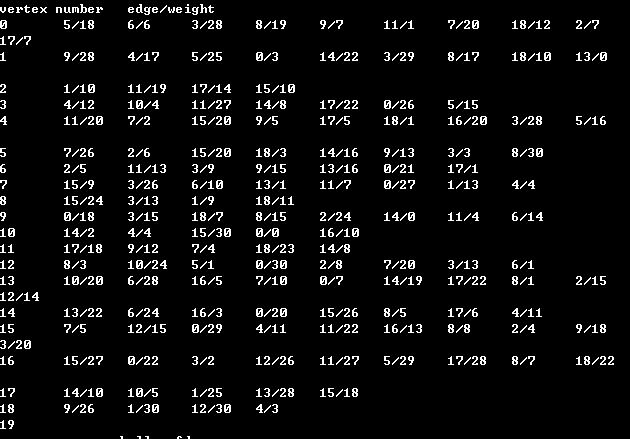




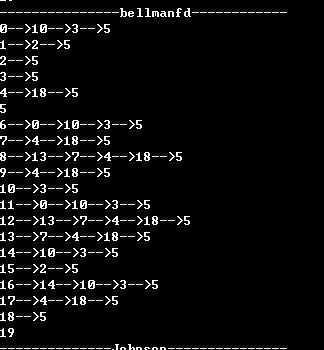


1. **运行结果**

打印图：



测试BF：



测试Johnson，打印d和pi表格，打印15->5的最短路径

