云南大学数学与统计学院
《算法图论实验》上机实践报告

课程名称：算法图论实验	**年级：**2015 级	**上机实践成绩：**
指导教师：李建平	**姓名：**	**专业：**
上机实践名称：编程实现求路的最小插点数	**学号：**20151910042	**上机实践日期：**2018-11-11
上机实践编号：4	**组号：**	

一、 实验目的

1. 了解最短路的最小插点问题的实际背景；
2. 能快速写出求最短路的最小插点问题的动态规划算法。

二、 实验内容

1. 写出求最短路的最小插点问题的动态规划算法；
2. 用 C 语言实现上述算法。

三、 实验平台

Windows 10 Pro 1803；

MacOS Mojave。

四、 算法设计

4.1 动态规划算法

本问题的核心算法是一个动态规划算法，此处花费一节来叙述动态规划算法的设计思路。

动态规划（Dynamic Programming）算法与分治策略类似，都是通过组合子问题的解来求解原问题。这是一种表格法。与分值策略不同的是，动态规划算法被应用于子问题重叠的情况，即不同的子问题具有公共的子子问题。动态规划算法会对每个子子问题只求解一次，将其保存在一个表格中，从而无需每次都求解相同的子子问题。这避免了相当的不必要的重复计算。

动态规划算法常用来求解最优化问题（Optimization Problem）。通常按照如下四个步骤来设计动态规划算法。

（1） 刻画一个最优解的结构特征；

（2） 递归地定义最优解的值；

（3） 计算最优解的值，通常是采取自底向上方法；

（4） 利用计算出的信息构造一个最优解。

最优子结构存在与子问题重叠是动态规划算法的两个特征。子问题的重叠使得缓存子问题的结果变得有意义，也使得降低计算量成为了可能。而最优子结构的存在，使得问题可以被比较简单地分治。通过研究矩阵连乘加括号问题，可以发现，如果以序列的形式输入一个含有n个矩阵的、相容的矩阵连乘：

（1） 设该矩阵连乘为$A_1A_2\cdots A_n$，记之为$A_{(1\to n)}$，类似地有子连乘式$A_{(i\to j)}=A_iA_{i+1}\cdots A_j$

（2） 该序列$\boldsymbol{p}$长度为$n+1$，即$\boldsymbol{p}=p_0p_1\cdots p_n$，它记录了整个连乘式子中所有矩阵的行列数信息；

（3） 连乘式中的第i个矩阵的行列指标反应在该序列中，$(r,\ c)=(p_{i-1},\ p_i)$；

（4） 定义$M_{(i\to j)}$被定义为计算$A_{(i\to j)}$所需要的最少标量乘法数；

下面指出一个极其重要的事实：在非平凡矩阵连乘的情况下，$M_{(1\to n)}=\min_{1\le k<n}\{M_{(1\to k)}+M_{(k+1\to n)}+p_0p_kp_n\}$。该事实指出，较长的序列的$M$值是由其子序列的$M$值决定的，所以可以使用自底向上计算，而且较长的子序列会包含较短的子序列，这就造成了子问题的重复出现。

4.2 最短路插点问题

路的插点问题与最短路的插点问题密切相关，这里先叙述一下“最短路插点”问题。

对于图$G=(V,\ E)$的所有节点对最短路径问题，可以证明一条最短路径的所有子路径都是最短路径。

定义 从节点u到节点v的最短路径权重$\delta(u,\ v)$如下：

$$\delta(u,\ v)=\begin{cases}\min\left\{w(p)\colon u\overset{p}{\Rightarrow}v\right\}, & \text{存在}u\text{到}v\text{的路}\\ \infty, & \text{其他}\end{cases}$$

问题描述：给定一副有向图$D=(V,\ E,\ w)$，$\forall u,\ s\in D.V$

五、 程序代码

六、 参考文献

[1] **林锐**. 高质量 C++/C 编程指南 [M]. 1.0 ed., 2001.

[2]