**2.2-何智鹏-Fast fully dynamic labelling for distence quries【 】**

**一、总结做法【 】**

本次演武堂汇报主题是：Fast fully dynamic labelling for distence quries。它是一种解决动态图的标签算法。

寻找任意一对顶点之间的最短路径距离是图论中的一个基本问题。大量的研究已经探索了这个问题，其中大部分都是局限在静态图。由于现实世界网络的动态性质，如社交网络或网络图，其中两个实体之间的链接可能在任何时候发生改变，目前迫切需要解决动态网络的这个问题。由于高空间成本和构建距离标签所需的长预处理时间，现有的工作只能在中等大的动态网络上适应距离查询。在这样中等大的动态网络中，距离标签很难有效地更新。在本文中，作者提出了一种全动态的标记方法来有效地更新距离标记，以此来回答大型动态图上的距离查询。在其核心算法上，作者提出的方法包含了两个构建块： (i)用于处理增量更新操作的增量算法，即边插入，和（ii）处理递减更新操作的递减算法，即边删除。这些构建块构建在一个高度可伸缩的距离查询回答框架中。

由于演武堂侧重在于实践动手和工程能力，因此需要对整个算法架构进行充分了解，拆解每一个部分。因此我花了大量的时间读代码，分析算法的实现步骤。

但是由于实践经验的欠缺，首先是对代码的上下逻辑关联性不是很理解，其次对代码里面很多的细节还不理解，比方说数据结构的组成，时间复杂度的分析。基于此，我查阅相关c++资料，了解一些涉及到的容器的使用，以及里面常用到的一些函数，先弄清楚这些常用函数的作用，然后再去看代码，进行debug调试，分析程序每一步运行的结果，最终理解了整个程序具体运行过程。

接着我还对数据集进行了处理：首先我向作者要了一份他的数据集，我将作者的数据进行了测试，测试的结果很符合论文里提到的效果。我还对论文当中给出的图例进行了评估，以通过实验验证作者的方法的效率、可扩展性和鲁棒性。

基于这些了解，我的演武堂PPT制作思路为五个部分：背景介绍、项目介绍、算法设计与实现、论文实验结果、未来工作展望。背景介绍主要讲了图论在我们实际生活当中的应用以及动态图最短路查询的相关研究。在算法设计与实现部分，为了更好的让听众理解，我制作了动画，给伪代码标注了执行步骤，像debug一样一步一步进行跳转来辅助大家理解。实验结果是根据作者的实验进行复现。未来工作展望中介绍了我们课题组接下来的工作。

**二、改进方法【 】**

**1.在汇报后现场老师们给我提出了一些建议：**

首先孙哲老师提出我的演武堂代码部分讲解的太少了，算法讲解的太多了，还是研讨厅的思维，需要改进。其次是企业老师点评，他的意见和孙哲老师保持了一致。在讲解的时候可以从项目的实现过程来解读，而不是以算法细节的方式。

**2.改进做法：**

根据老师的批评意见，我进行了总结，从以下几点做改进：

1. 代码片段：选取代码中的关键部分或有代表性的部分，将其放入代码片段中进行展示。同时，可以通过注释或者解释文字来说明代码的功能和作用。

2. 代码流程图：将代码的逻辑流程图形化展示出来，这可以帮助观众更好地理解代码的执行流程。代码流程图可以使用工具如Microsoft Visio、Lucidchart、Draw.io等进行绘制。

3. 伪代码：将代码转换为伪代码，将重点关注算法的核心思想和执行流程，而不必太过注重具体的编程语言和细节。伪代码可以使观众更容易理解代码的算法思路。但不能展示过多的伪代码。

4. 示例程序：编写简化版的程序，用于展示算法的核心思想和流程，而不必涉及太多细节。示例程序可以使观众更好地理解算法的实现过程。

在展示代码的过程中，还应注意以下几点：简明扼要：尽量简化代码，去除不必要的部分，保留最核心的部分，以便更好地说明算法的思路和流程。呈现清晰：代码应该以清晰、易读的方式呈现，可以调整字体大小、颜色等方式使代码更加易于阅读。注重说明：除了展示代码，还应该对代码进行解释说明，以便让观众更好地理解算法的思路和流程。

综上所述，以上方式都可以使观众更好地理解论文的算法思路和流程，达到演武堂对工程的锻炼能力，提升自己的代码能力。