# 实验报告模板

华南理工大学

《课程名称》课程实验报告

实验题目：**DP shortest path algorithm**

姓名：ZULKARNINE MOHAMMED SHAHLA,

CHRISTOPHER GABRIEL PINHIERO,

GLIGOROVSKI NIKOLA,

TADESSE ABEL MASRESHA

学号：201869990129,

 201869990420,

 201869990252,

 201869990725

班级：International students

指导教师：Yan Xiaoyang

|  |
| --- |
| **实验概述** |
| 【Purposes and requirements】  Purpose: Compare  Requirements:  【lab environment】  OS：Windows 10 |
| **实验内容** |
| 【experiment procedure】  1st Experimental step: Step 1: Initialize distances from src to all other vertices as INFINITE  Step 2: Relax all edges |V| - 1 times. A simple shortest  path from src to any other vertex can have at-most |V| - 1 edges  Step 3: check for negative-weight cycles. The above step guarantees shortest distances if graph doesn't contain negative weight cycle. If we get a shorter path, then there is a cycle.  2nd Experimental data: Let us understand the algorithm with following example graph. The images are taken from [this](http://www.cs.arizona.edu/classes/cs445/spring07/ShortestPath2.prn.pdf)source.  Let the given source vertex be 0. Initialize all distances as infinite, except the distance to the source itself. Total number of vertices in the graph is 5, so *all edges must be processed 4 times. Let all edges are processed in the following order: (B, E), (D, B), (B, D), (A, B), (A, C), (D, C), (B, C), (E, D). We get the following distances when all edges are processed the first time. The first row shows initial distances. The second row shows distances when edges (B, E), (D, B), (B, D) and (A, B) are processed. The third row shows distances when (A, C) is processed. The fourth row shows when (D, C), (B, C) and (E, D) are processed.* The first iteration guarantees to give all shortest paths which are at most 1 edge long. We get the following distances when all edges are processed second time (The last row shows final values). The second iteration guarantees to give all shortest paths which are at most 2 edges long. The algorithm processes all edges 2 more times. The distances are minimized after the second iteration, so third and fourth iterations don’t update the distances.  **Notes** **1)**Negative weights are found in various applications of graphs. For example, instead of paying cost for a path, we may get some advantage if we follow the path.  3rd The main process of the experiment: Like other Dynamic Programming Problems, the algorithm calculates shortest paths in a bottom-up manner. It first calculates the shortest distances which have at-most one edge in the path. Then, it calculates the shortest paths with at-most 2 edges, and so on. After the i-th iteration of the outer loop, the shortest paths with at most i edges are calculated. There can be maximum |V| – 1 edges in any simple path, that is why the outer loop runs |v| – 1 times. The idea is, assuming that there is no negative weight cycle, if we have calculated shortest paths with at most i edges, then an iteration over all edges guarantees to give shortest path with at-most (i+1) edges |
| **小结 conclusion** |
| **Concluded above.** |
| **指导教师评语及成绩** |
| 评语：  成绩：           指导教师签名：                                                 批阅日期： |