CSIT 503 HW8
Topic: Single Source Shortest Path

Dr. Boxiang Dong

1 Problem Description

Instructions. You are provided one skeleton program named *Graph2.java*. The source files are available on Canvas in a folder named *HW8*. Please modify the skeleton code to solve the following tasks.

- Task 1 (50 pts). Implement the *bellman_ford(int s)* function as discussed in Lecture 19.

  Note: You **should** return an boolean value.

- Task 2 (50 pts). Implement the *dijkstra(int s)* function as discussed in Lecture 20.

  - **Hint 1:** We use an adjacent matrix to represent the graph. If $A[i][j] = 0$, it means there is no edge between the i-th and j-th node. If $A[i][j] \neq 0$, then it means the weight of the edge between the i-th and j-th node.
  - **Hint 2:** You do not need to do any operation for the π variable in the pseudocode in Task 1 and Task 2.

- Task 3 (50 pts (Extra Credit)). Implement a function to organize the shortest path for each node as a string. For example, if a node 4's shortest path is $0 \rightarrow 2 \rightarrow 1 \rightarrow 4$, you can generate a string variable $s =$"$0 \rightarrow 2 \rightarrow 1 \rightarrow 4$". Modify the *display_distance()* function to show the shortest distance and the shortest path for each node.

  - **Hint 1:** You definitely need to do operation for the π variable in this task. Feel free to add any class member data based on your need.

2 Submission Guideline

1. Work individually.
2. Please directly insert your code in the appropriate place in Graph3.java.
3. Create a zip file of your .java source programs and submit it on Canvas on time. A late penalty of 10 points for each late day applies. Any late for more than three days receives zero automatically.