Lab 8: Shortest Paths

COSC 3020: Algorithms and Data Structures

Lars Kotthoff
larsko@uwyo.edu

06 November 2019

Instructions

Attempt to finish the tasks below during the lab time. You have until Friday, 08 November 2019, 23:59h to submit the solutions to WyoCourses. You may ask your TA for feedback before submitting, but this feedback will be qualitative only.

You may *not* use external libraries in your code unless explicitly stated.

1 All-Pairs Shortest Paths

In the lectures, we've seen Dijkstra's algorithm for finding the shortest paths from a given vertex to all other vertices in the graph. The Floyd-Warshall algorithm for finding the shortest path between all *pairs* of vertices works as follows:

Given a graph $G = (V, E)$ with weighted edges:

```
• initialize a $|V| \times |V|$ matrix dist to $\infty$
• for each vertex $v \in V$, dist[v][v] = 0
• for each edge $(u, v) = e \in E$, dist[u][v] = weight((u,v))
• for each vertex $k \in V$:
    – for each vertex $i \in V$:
        ∗ for each vertex $j \in V$:
            · if dist[i][j] > dist[i][k] + dist[k][j]:
                  dist[i][j] = dist[i][k] + dist[k][j]
```

Implement the function `allPairsShortestPaths` that takes a weighted graph and returns the matrix with the distances, as described above. Test it on a few different graphs, ideally with an automated testing framework. You can choose any data structures you like for the implementation.

What is the worst-case time complexity (Θ) of the algorithm?

Total 10 points.