Lab-8

DA5300: Data Structures for Data Science

October 23, 2024

Let $V = \{0, \ldots, n-1\}$ be the set of vertices of a directed graph. Let A be the $n \times n$ adjacency matrix (an 2 dimensional `numpy.array` of shape (n,n)).

1) Write a function `Bellman-Ford(A,s)` which accepts the adjacency matrix A, and the start vertex s. The function should return `cycle,delta,Pi`. The function should return `cycle=0` if there is no negative cycle and `cycle=1` if there is a negative cycle. `delta` is an array of length n, with `delta[i]` is the length of the shortest path from `s` to `i`. `Pi` is an array of length n, with `Pi[i]` being the predecessor of `i` in the shortest path from `s`.

2) Assume all the edge costs to be positive. Similar to previous question, write a function `Dijkstra(A,s)` which returns `delta,Pi`.

3) Assume that no negative cycles exists. Write a function `Floyd-Warshall(A)` which accepts the adjacency matrix A. The function should return `delta,Pi`. `delta` is an numpy array of shape (n,n), with `delta[i,j]` is the length of the shortest path from `i` to `j`. `Pi` is an numpy array of shape (n,n), with `Pi[i][j]` being the predecessor of `j` in the shortest path from `i`.