CSCI 102 assignment 8 – Graph Traversals

April 29, 2024

The traveling salesman problems asks if there is a path less than length `target` in an undirected graph that visits every node **at most once**.

- Implement a method `static <V> GoodList<Edge<Double>> travellingSalesman(Graph<V, Double> graph, Vertex<V> start, double target)` that solves the travelling salesman problem, considering all paths starting at `vertex<V>`. This method should use recursive backtracking.
- Assuming `outgoingEdges` is $O(1)$, what is the worst case asymptotic computational complexity of this method in terms of the number of vertices n, number of edges m, and maximum degree d? Hint: it's large.
- Using any graph implementation you like, create a graph with 5 vertices labelled 1 through five such that all vertices are connected by forward and backward edges except there is no edge between 3 and 5 and between 1 and 4. Weight all edges by the average of the opposite vertices – i.e. the weight of the edge from 3 to 5 should be 4 – except the edge between 1 and 2 should have weight 1. Apply your method to solve the travelling salesman problem to this graph with targets $0.5, 1, 1.5, \ldots, 9.5, 10$ (that's 20 calls to your method in total).

Please submit your code and answers to the questions in a zipped folder on Brightspace by April 29.