Greedy Algorithm Drasil Case Study

Don Chen

March 16, 2021

This is the proposal of the family of greedy algorithms to my drasil case study.

1 Problem Description

- Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once?

2 Goal Statements

- General: Find the shortest possible route that visits each city exactly once.
- Mathematical Definition: In the weighted graph, find the shorest path which contains all vertices.

3 Assumptions

- V and E are finite sets.
- It is undirected graphs, means f(e) = uv = vu.
- It is a connected graph.
- There is no loop in the graph.

4 Theoretical Model

- Nearest Neighbor Alorithm
- Kruskal's Algorithm

5 Defined In Graphic

- Each city is a node, and each node will has its name.
- A node can connect to other nodes with a line.
- Once two nodes connected, there will be a distance assgined on the line.

6 General Definition

A graph G is an ordered triple (V(G), E(G), f) consisting of a nonempty set V(G) of vertices, a set E(G) of edges, and an incidence function f that associates with each edge and two vertices. If e is an edge and u and v are vertices, u, v $\in$ V (G), such that f(e) = uv, then e is said to join u and v; the vertices u and v are called the ends of e. u and v are endpoints of edge e. A weighted graph is a graph with each edge e of G let there be associated a real number w(e).

- G = (V(G), E(G), f, w)
- V(G) = v_1, v_2, v_3, v_4, v_5
- E(G) = e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8
- f(e_1) = v_1v_2, f(e_2) = v_2v_3, f(e3) = v_3v_3, f(e4) = v_3v_4,
- f(e5) = v_2v_4, f(e_6) = v_4v_5, f(e_7) = v_2v_5, f(e_8) = v_2v_5
- w(e_1) = n_1, w(e_2) = n_2, w(e_3) = n_3, w(e_4) = n_4,
- w(e_5) = n_5, w(e_6) = n_6, w(e_7) = n_7, w(e_8) = n_8

7 Supporting Data Definitions

- Finite Graph: A graph is finite if both its vertex set and edge set are finite.
- Endpoint: if f(e) = uv, edge e has endpints of u and v. u and v $\in$ V(G).
- Loop: An edge with identical ends. f(e) = vu and v = u. u and v $\in$ V(G).
- Connected Graph: if for every u, v $\in$ V(G) there exists f(e) = uv. Otherwise G is called disconnected.

- Walk: A walk is a sequence W = v_0, e_1, v_1, ..., e_k, v_k, whose terms are alternatively vertices and edges such that for 1≤i≤k, the edge e_i has endpoints v_{i-1} and v_i. k is the length of W.
- Trail: If the edges e_1, e_2, ..., e_k of a walk are distinct.
- Path: a path is a walk if both of its edges e_1, e_2, ..., e_k and its vertices v_0, v_1, ... , v_k are distinct.
- Closed: A u,v-walk has first vertex u and last vertex v. When the first and last vertex of a walk, trail or path are the same, we say that they are closed.
- The Shorest Path: a path has the smallest number of sum of its edges' weight.
- Weight: With each edge e of G let there be associated a real number w(e), called its weight.

8 Instance Models

- Input: G = (V(G), E(G), f, w)
- Output: a sequence

9 Question

- clarify on Walk, how the sequence formed? V and G are set in the definition.
- remove the conditon of "returns to the origin city"

10 Reference

- Graph Theory With Applications by J.A. Bondy and U. S. R. Murty
- Nearest Neighbor Alorithm
- Kruskal Alorithm in TSP