Problem C: The GbAaY Kingdom

Brief

Program Name: `PT07C`
Method Summary: Find the minimum diameter spanning tree
Time Complexity: $O(n \cdot m)$

Editorial

Only three contestants solved this problem. **mountainking** (Lou Tiancheng) is the first one.

The standard algorithm is $O(n \cdot m)$. The $O(n \cdot m \cdot \log n)$ algorithm is hard to get accepted.

Solution

This problem is a classical problem, called the Minimum Diameter Spanning Tree. The $O(n \cdot m)$ algorithm [5] will be introduced in the following.

Notation C.1 (Basic Graph Glossary) Let $G = (V, E)$ be a connected, undirected, positively real-weighted graph, where the weight of an edge $e = (u, v) \in E$ is given by $\omega_{u,v} \in \mathbb{R}_+$.

- For all vertices u and v, the *distance* from u to v, denoted $d_G(u, v)$, is the shortest path from u to v in G. We call all $d_G(u, v)$s *all-pairs shortest paths* (APSP) of G. Sometimes, $d_G(u, v)$ abbreviates $d(u, v)$ for G.
- The maximal distance from vertex v to all other vertices in V, denoted $\varepsilon(v)$, is the *eccentricity* of v, i.e. $\varepsilon(v) = \max_{u \in V} d(v, u)$.
- $D(G)$ denotes the *diameter* of G, defined as $D(G) = \max_{v \in V} \varepsilon(v)$.
- The *minimum diameter spanning tree* (MDST) of G is a spanning tree T of G minimizing $D(T)$.
- $R(G)$ denotes the *radius* of G, defined as $R(G) = \min_{v \in V} \varepsilon(v)$.
- $\Psi(u)$ represents a *shortest-paths tree* (SPT) rooted at node u, such that $\forall v \in V, d(u, v) = d_{\Psi(u)}(u, v)$. ■

[5] Marc Bui, Franck Butelley, Christian Lavaulty, *A Distributed Algorithm for the Minimum Diameter Spanning Tree Problem*

Suppose that the median of the longest path of the MDST of G is on a certain vertex $v \in V$. Then the SPT $\Psi(v)$ will be a MDST of G. Thus we can enumerate all $v \in V$ and find the SPT $\Psi(v)$ having the minimum diameter $D(\Psi(v))$.

However, in fact, the median of the longest path of the MDST of G can appear in inner of a certain edge. Thus, we suggest the "dummy vertex" (so-called in contrast to actual vertices of V), that is a fictitious vertex may possibly be inserted on any edge $e \in E$.

Definition C.2 *Let $e = (u, v)$ be an edge of weight $\omega_{u,v}$.*

- *A **dummy vertex** γ inserted on e is defined by specifying the weight α of the "dummy edge" (u, γ).*
- *A **general vertex** is an actual vertex in V or a dummy vertex.*

Due to the generalization of the vertex, we should generalize all the concepts in Notation C.1. The domains of the distance, the eccentricity and the SPT are generalized to the general vertices, i.e. $d(\gamma, \gamma'), \varepsilon(\gamma), \Psi(\gamma)$.

If we could enumerate all general vertices in G, the MDST of G would be the SPT $\Psi(\gamma)$ having the minimum diameter. However, because there are infinite general vertices in G, we can't enumerate all of them. Nevertheless, in other way, we will find the "absolute center" of G directly.

Definition C.3 *An **absolute center** γ^* of G is defined as a general vertex γ having the minimal eccentricity $\varepsilon(\gamma)$.*

The absolute center γ^* of G is also the median of a longest path of the MDST $\Psi(\gamma^*)$. Obviously, the MDST problem for a graph G is reducible to the absolute center problem. The following shows how to find the absolute center.

1. For each edge $e \in E$, find a general vertex $\gamma \in e$ having the minimum eccentricity $\varepsilon(\gamma)$, denoted by γ_e.
2. γ^* is a general vertex having the minimum eccentricity among all the above γ_es.

The remaining work is to compute γ_e of a particular edge e.

For edge $e = (u, v) \in E$, let $\alpha = d(u, \gamma)$. Because for each other vertex w, the distance $d(\gamma, w)$ is the length of either a path $\gamma, u, \ldots, w$, or $\gamma, v, \ldots, w$. Hence,

$$\varepsilon(\gamma) = \max_{w \in V} d(\gamma, w) = \max_{w \in V} \min\{\alpha + d(u, w), \omega_{u,v} - \alpha + d(v, w)\}$$

If we plot $d(\gamma, w)$ in Cartesian coordinates for fixed w, the curve of $d(\gamma, w)$ are represented by two line segments with the slope $+1$ and -1. The curve of $d(\gamma, w)$ can be determined by only two numbers $d(u, w)$, $d(v, w)$, denoted by $a_w = d(u, w)$, $b_w = d(v, w)$ respectively. If we plot $d(\gamma, w)$ for all w, the curve of $\varepsilon(\gamma)$ are represented by a broken line that is the upper boundary of the convex cone of all $d(\gamma, w)$. The vertex γ achieving the global minimum value of the broken line represents the absolute center γ_e of the edge e. See Figure 5 illustratively.

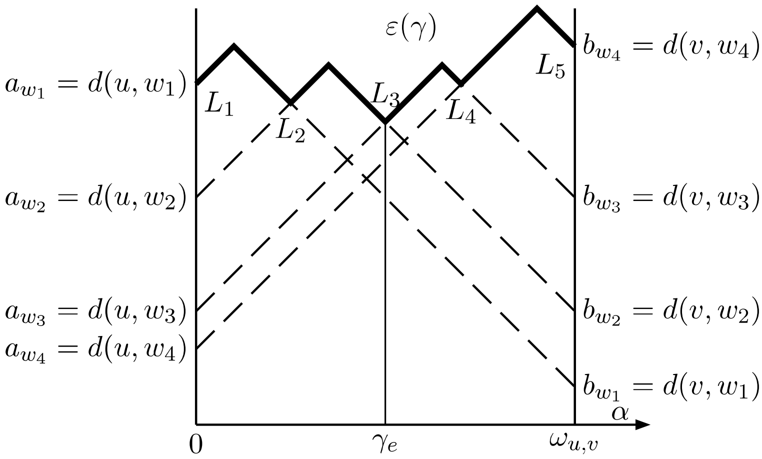

Figure 5: The curve of $\varepsilon(\gamma)$.

If there exists $d(u, w) \leq d(u, w')$ and $d(v, w) \leq d(v, w')$ for two certain vertices w, w', the curve of $d(\gamma, w)$ will be completely under the curve of $d(\gamma, w')$, and then we can remove w and w's curve. We call removing all above-mentioned useless vertices w by "simplification". If we plot $d(u, w) = a_w$, $d(v, w) = b_w$ as a coordinate (a_w, b_w) for all $w \in V$ (See Figure 6), we will find that only the upper boundary of the convex cone of the points can be remained after "simplification".

After "simplification", suppose w has already been ordered by a_w, the local minimum value L_w of $\varepsilon(\gamma)$ will be intersected by the curves of $d(\gamma, w)$ of two adjacent ws (e.g. In Figure 5, L_2 is intersected by the curves of $d(\gamma, w_1)$ and $d(\gamma, w_2)$). We can the global minimum value γ_e among all local minimum values L_w.

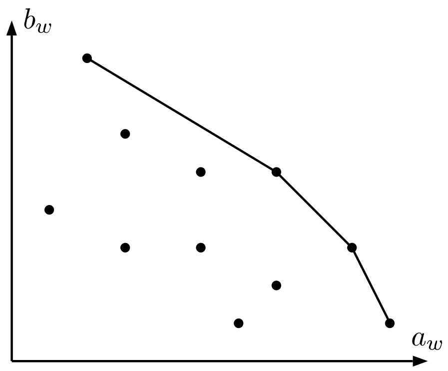

Figure 6: The simplification of w.

Algorithm 2 MDST

Require: Given a connected, undirected, positively real-weighted graph $G = (V, E)$
Ensure: Return the MDST of G

```
function MDST(G)
    run Floyd-Warshall(f)or finding all d(u,v) (APSP)
    for all e = (u,v) ∈ E do
        Phase 1: Simplification in O(n) time
        S = ∅                                   ▷ S stores all w remained after "simplification"
        for all w ∈ V ordered by a_w = d(u,w) do
            update S by w (a_w, b_w)
        end for

        Phase 2: Compute γ_e in O(n) time
        for all w ∈ S ordered by a_w = d(u,w) do
            compute the local minimum value L_w intersected by the curves of w and w+1
            update γ_e by L_w
        end for
        update γ* by γ_e
    end for
    return Ψ(γ*)
end function
```

The time complexity of the algorithm above is obviously $O(n \cdot m)$.