Analysis of Dijkstra's Algorithm

Hu Tianrun

September 14, 2021

1 Definitions, Assumptions and Lemmas

1.1 Shortest-path weight definition

In a **shortest-paths problem**, we are given a weighted, directed graph $G = (V, E)$, with function $w = E \to \mathbb{R}$ mapping edges to real-valued weights.

We define the **shortest-path weight** $\delta(u, v)$ from u to v,

$$\delta(u,v) = \begin{cases} \min\{w(p) : u \overset{p}{\leadsto} v\} & \text{if there is a path from } u \text{ to } v, \\ \infty & \text{otherwise.} \end{cases}$$

1.2 Optimal substructure of a shortest path

Lemma Given a weighted, directed graph $G = (V, E)$ with weight function $w = E \to \mathbb{R}$, let $p = \langle v_0, v_1, \ldots, v_k \rangle$ be a shortest path from vertex v_0 to vertex v_k and, for any i and j such that $0 \leq i \leq j \leq k$, let $p_{ij} = \langle v_i, v_{i+1}, \ldots, v_j \rangle$ be the subpath of p from vertex v_i to vertex v_j. Then, p_{ij} is a shortest path from v_i to v_j.

Proof If p_{ij} is not the shortest path, we can replace it by shortest path to build a shorter path from v_0 to v_k which lead to a contradiction.

1.3 Representing shortest paths

Given a graph $G = (V, E)$, we build a new array *parent* to note the parent for each vertex in G. Thus, we can define a function printPath(s, v) to print the shortest path from s to v.