Dijkstras Algorithms

Arnav Gupta

April 9, 2024

Contents

1 Preliminaries

A graph $G = (V, E)$ is a directed graph with a weight function: $w : E \to \mathbb{R}$

- weight of the path $P = \langle v_0, \ldots, v_k \rangle$ is $w(P) = \sum_{i=1}^{k} w(v_{i-1}, v_i)$

Shortest path does not exist for directed weighted graphs with negative-weight cycles.

Under the assumption that G has no negative-weight cycles, the shortest path weight from u to v:

$$\delta(u, v) = \begin{cases} \min\{w(P) : u \to v\} & \text{if there exists a path from } u \text{ to } v \\ \infty & \text{otherwise} \end{cases}$$

1.1 Single-Source Shortest Path Problem

- input: $G = (V, E)$, $w : E \to \mathbb{R}$ and a source $s \in v$
- output: a shortest path from s to each $v \in V$

Consider the following: if $\langle v_0, v_1, \ldots, v_k \rangle$ is a shortest path from v_0 to v_k, then $\langle v_0, v_1, \ldots, v_k \rangle$ is a shortest path from v_0 to v_i, for any $0 \leq i \leq k$

2 Dijkstra's Algorithm

A greedy algorithm that takes a weighted directed graph with non-negative edge weights.

Important quantities:

- $d[v]$: a shortest path estimate from s to v
- $\pi[v]$: predecessor in the path (a vertex or NIL)

2.1 Explanation

- iniitialize $C = \emptyset$, repeat the following untit $C = V$
    1. add $u \in V - C$ with smallest d value to C
    2. update d values of vertices v with $(u, v) \in E$:

       $$d[v] \leftarrow \min\{d[v], d[u] + w(u, v)\}$$

    3. update $\pi[v]$ if $d[v]$ is changed
- Priority Queue is ADT that should be used for vertices
    - implemented as binary min-heap with costs
        * insert: $O(\log(n))$
        * extract-min: $O(\log(n))$
        * update-key: $O(\log(n))$

2.2 Complexity Analysis

- array implementation has time complexity $O(|V|^2)$
- heap implementation has time complexity $O((|V| + |E|)\log(|V|))$