CS 577: HW 3

Daniel Ko

Summer 2020

§1 Kleinberg Chapter 6, Q14

§1.1 Suppose it is possible to choose a single path P that is an $s-t$ path in each of the graphs $G_0, G_1, \cdots, G_b$. Give a polynomial-time algorithm to find the shortest such path.

§1.1.1 Set up the recursive formula and justify its correctness.

We define the set of edges that exist in all points in time as

$$E^* = \bigcap_{i=0}^{b} E_i$$

Similiar to (6.23) from the book, we can modify the Bellman–Ford algorithm to find the shortest path from s to t. We define $\mathrm{OPT}(i, v)$ to be the length of the shortest path from v to t using at most i edges.

$$\mathrm{OPT}(i, v) = \min(\mathrm{OPT}(i-1, v), \min(\{1 + \mathrm{OPT}(i-1, w) \mid (v, w) \in E^*\}))$$

Proof. idk man $\square$

§1.1.2 Write the pseudocode for the iterative version of the algorithm to find the minimum cost. You are not required to write pseudocode to find the shortest path.

§1.1.3 Analyze the computing complexity.

§1.2 Give a polynomial-time algorithm to find a sequence of paths $P_0, P_1, \cdots, P_b$ of minimum cost, where P_i is an $s-t$ path in G_i for $i = 0, 1, \cdots, b$.