Written Assignment 1 — Solutions

CS 440 September 17, 2025

Name: Yat Long Szeto BU ID: 90479281

Question 1: Shortest Path Composition

Proof

$$\begin{aligned}
p^* &= p_1 \cdot p_2 && \text{split } a \to b \text{ path at } c \\
\text{cost}(p^*) &= \text{cost}(p_1) + \text{cost}(p_2) && \text{path cost additivity} \\
\exists\, \tilde{p}_1 :\ \text{cost}(\tilde{p}_1) &< \text{cost}(p_1) && \text{assume } p_1 \text{ not shortest} \\
\text{cost}(\tilde{p}_1 \cdot p_2) &= \text{cost}(\tilde{p}_1) + \text{cost}(p_2) && \text{concatenate paths} \\
&< \text{cost}(p_1) + \text{cost}(p_2) && \text{by assumption} \\
&= \text{cost}(p^*) && \text{substitution}
\end{aligned}$$

This contradicts the optimality of p^*. Therefore p_1 must be a shortest $a \to c$ path and there is no such $\tilde{p}_1$ exists. WLOG, replace the prefix p_1 by the suffix p_2 and c by b. The same reasoning shows that if p_2 were not shortest $c \to b$, then replacing it with a shorter path would also contradict the optimality of p^*. Hence p_2 is a shortest $c \to b$ path as well.

Question 2: Iterative Deepening Returns Shortest Paths

Proof

$$\begin{aligned}
\ell(v) &= \min\{\text{length}(P) : P \text{ is an } s \to v \text{ path}\} && \text{define shortest distance} \\
\text{When IDDFS, } d < \ell(v) &\ \Rightarrow\ v \text{ not found} && \text{too shallow} \\
d = \ell(v) &\ \Rightarrow\ v \text{ reachable} && \text{path length fits} \\
\text{IDDFS finishes depth } d &= \ell(v) \text{ before } d+1 && \text{order of search} \\
v \text{ first discovered} &= \text{ at depth } \ell(v) && \text{cannot appear earlier}
\end{aligned}$$

Therefore, the first time v is returned by Iterative Deepening is exactly at depth $\ell(v)$, and the path has the minimum number of edges.

Question 3: Diameter Bound

Proof

$$
\begin{array}{lr}
\text{Fix } u, v \in V,\ u \neq v. & \text{setup} \\
W = (v_0, \ldots, v_k),\ v_0 = u,\ v_k = v & \text{a } u \to v \text{ walk of minimum length} \\
\exists\, i < j:\ v_i = v_j \ \Rightarrow\ W' = (v_0, \ldots, v_i, v_{j+1}, \ldots, v_k) & \text{delete cycle} \\
\operatorname{len}(W') = k - (j - i) < k = \operatorname{len}(W) & \text{strictly shorter} \\
\Rightarrow\ \neg\exists\, i < j:\ v_i = v_j & \text{contradiction to minimality} \\
\Rightarrow\ W \text{ is simple} & \text{no repetitions} \\
\text{Let } P = W,\ |P| = m \text{ (edges)} \ \Rightarrow\ P \text{ visits } m+1 \text{ distinct vertices} & \text{path has one more vertex} \\
m + 1 \leq |V| \ \Rightarrow\ m \leq |V| - 1 & \text{counting bound} \\
\operatorname{dist}(u, v) = |P| \leq |V| - 1 & \text{shortest path equals } |P| \\
\operatorname{diam}(G) = \max\limits_{u \neq v} \operatorname{dist}(u, v) \leq |V| - 1 & \text{take max}
\end{array}
$$

Question 4 — Dijkstra's Algorithm and Shortest Paths

Theorem 1. *Let $G = (V, E, w)$ be a directed graph with strictly positive edge weights $w(e) > 0$. When Dijkstra's algorithm is run from a source s and it returns a path to any vertex v, the returned path has length $\delta(s, v)$, the true shortest-path distance from s to v.*

Proof. Assume for contradiction that some vertex is *settled* with an incorrect label. Let v be the first such vertex extracted from the priority queue with $d[v] > \delta(s, v)$; thus, every previously settled u satisfies $d[u] = \delta(s, u)$.

Consider a shortest $s \to v$ path P, and let y be the first vertex on P that is not yet settled just before v is extracted; let x be the predecessor of y on P. By choice of y, x is settled. By the induction hypothesis for earlier settled vertices, $d[x] = \delta(s, x)$. When x was settled, the relaxation of edge (x, y) gave

$$d[y] \ \leq\ d[x] + w(x, y) \ =\ \delta(s, x) + w(x, y) \ =\ \delta(s, y),$$

and since y lies on a shortest $s \to v$ path, we have $\delta(s, y) \leq \delta(s, v)$. Hence

$$d[y] \ \leq\ \delta(s, v).$$

Because Dijkstra extracts the unsettled vertex with minimum key, it holds that

$$d[v] \ \leq\ d[y] \ \leq\ \delta(s, v),$$

contradicting the assumption $d[v] > \delta(s, v)$. Therefore no such v exists, and every vertex is settled with its true distance; in particular, the path returned to any v is a shortest path. $\square$