Written Assignment 1 — Solutions

CS 440 September 16, 2025

Name: Yat Long Szeto BU ID: 90479281

Question 1: Shortest Path Composition

Answer

$$\begin{aligned}
p^* &= p_1 \cdot p_2 && \text{split the } a \to b \text{ path at } c \text{ (concatenation)} \\
\text{cost}(p^*) &= \text{cost}(p_1) + \text{cost}(p_2) && \text{path cost additivity} \\
\exists\, \tilde{p}_1 :\ \text{cost}(\tilde{p}_1) &< \text{cost}(p_1) && \text{assumption for contradiction} \\
\text{cost}(\tilde{p}_1 \cdot p_2) &= \text{cost}(\tilde{p}_1) + \text{cost}(p_2) && \text{additivity} \\
&< \text{cost}(p_1) + \text{cost}(p_2) && \text{by assumption} \\
&= \text{cost}(p^*) && \text{from the first two lines}
\end{aligned}$$

Since $\text{cost}(\tilde{p}_1 \cdot p_2) < \text{cost}(p^*)$, this contradicts that p^* is a shortest $a \to b$ path. Hence p_1 must be shortest $a \to c$. By the same argument, p_2 is shortest $c \to b$.

Question 2: Iterative Deepening returns shortest paths

Theorem 1. *On an unweighted (unit-edge) graph, Iterative Deepening DFS with limits $1, 2, 3, \ldots$ from a source s returns, for any discovered vertex v, a path with the minimum number of edges.*

Proof. Let $\ell(v)$ be the shortest-path distance (in edges) from s to v. For $d < \ell(v)$, depth-limited DFS cannot expose any s–v path. When $d = \ell(v)$, a shortest s–v path of length $\ell(v)$ is eligible and will be found during that pass; the procedure completes the whole $d = \ell(v)$ pass before running $d = \ell(v)+1$. Therefore the first time v is returned is at depth limit $\ell(v)$, with a shortest path. □

Question 3: Diameter bound

Proof. For any $u \neq v$, any simple u–v path visits vertices at most once, hence uses at most $|V| - 1$ edges. A shortest u–v path is simple (otherwise remove a cycle to shorten it). So $\text{dist}(u, v) \leq |V| - 1$ for all pairs, and the maximum over pairs is at most $|V| - 1$. The diameter is at most $|V| - 1$. □

Question 4 — Dijkstra's Algorithm and Shortest Paths

Theorem 2. *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. $\square$