Written Assignment 1 — Solutions

CS 440 September 16, 2025

Name: Yat Long Szeto BU ID: 90479281

Question 1: Shortest Path Composition

Answer

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

Since $\operatorname{cost}(\tilde{p}_1 \cdot p_2) < \operatorname{cost}(p^*)$, 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.*

Answer

$$
\begin{aligned}
\ell(v) &= \min\{\operatorname{length}(P) : P \text{ is an } s \to v \text{ path}\} && \text{define } \ell(v) \text{ as shortest dist} \\
d < \ell(v) &\ \Rightarrow\ \text{no } s \to v \text{ path found} && \text{depth bound too s} \\
d = \ell(v) &\ \Rightarrow\ \text{a shortest } s \to v \text{ path is eligible} && \text{length } \ell(v) \text{ now allo} \\
\text{algorithm completes all of depth } d = \ell(v) &\text{ before } d = \ell(v) + 1 && \text{order of iterative deepe} \\
\Rightarrow\ v \text{ first discovered at depth } &= \ell(v) && \text{cannot appear earlier, appears} \\
\text{thus path returned has length } &= \ell(v) && \text{definition of shortest path le}
\end{aligned}
$$

Question 3: Diameter bound

Theorem 2. *For any finite, undirected, unweighted graph $G = (V, E)$, the diameter satisfies* $\operatorname{diam}(G) \le |V| - 1$.

Proof. Fix $u \ne v$. 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 $\operatorname{dist}(u, v) \le |V| - 1$ for all pairs, and the maximum over pairs is at most $|V| - 1$. □

Question 4: Correctness of Dijkstra's algorithm (positive weights)

Theorem 3. *Let $G = (V, E, w)$ be directed with $w(e) > 0$. Dijkstra's algorithm from source s settles each vertex v with key $d[v] = \delta(s, v)$, the true shortest-path distance.*

Loop invariant proof. Invariant. (i) For every settled u, $d[u] = \delta(s, u)$. (ii) For every fringe x, $d[x]$ equals the minimum length of an s–x path whose last edge enters from a settled predecessor.

Initially only s may be settled with $d[s] = 0 = \delta(s, s)$; relaxations preserve (ii). Suppose before an iteration the invariant holds and the algorithm extracts v with minimal $d[v]$ among unsettled vertices. If $d[v] > \delta(s, v)$, take a shortest s–v path and let y be the first unsettled vertex on it with settled predecessor x. Then $d[x] = \delta(s, x)$ by (i), and relaxing (x, y) gave $d[y] \le \delta(s, y) \le \delta(s, v)$. Minimality of $d[v]$ implies $d[v] \le d[y] \le \delta(s, v)$, a contradiction. Hence $d[v] = \delta(s, v)$ when settled; relaxations maintain (ii). By induction, the claim holds for all settled vertices. □

Extra Credit (outline): Column-constrained top-to-bottom shortest path on a vertex-weighted lattice

Model. $n \times n$ directed lattice; edges have weight 0; each vertex u has weight $w(u) > 0$; path length is the sum of vertex weights (including the start).

Goal. For each column i, let P_i be a shortest path from the top vertex $v_{1,i}$ to bottom vertex $v_{n,i}$. Output $P^* = \arg\min_i \operatorname{len}(P_i)$.

Reduction using the given oracle. Augment G with super-source s and super-sink t of zero weight. Connect $s \to v_{1,i}$ for all i. To enforce that the path ends in the *same* column, build, for each fixed i, a graph $G^{(i)}$ that includes only the edge $v_{n,i} \to t$ (delete $v_{n,j} \to t$ for $j \ne i$). One call to the oracle on $(G^{(i)}, s, t)$ returns P_i. Take the best over i. This takes $O(n)$ oracle calls on $\Theta(n^2)$-size lattices (overall $O(n^3)$). If lateral moves cannot change the column at the bottom, a single call on the graph with all $v_{n,i} \to t$ suffices in $O(n^2)$.

Correctness. In $G^{(i)}$, any s–t path must start at some $v_{1,i}$ and end at $v_{n,i}$; its cost equals the vertex-sum along the $v_{1,i} \rightsquigarrow v_{n,i}$ segment, i.e., the length of P_i. Minimizing over i yields P^*.