CS 3510 — Algorithms, Spring 2022

Homework 3A Solutions

1 Restricted Paths

Description. First, we want to compute the distance from v_0 to every other node. This can be done using Dijkstra. Hence, we run $\texttt{Dijkstra}(G = (V, E), v_0)$ to get a distance ($dist_{fwd}$) map from $v_0 \to v : \forall v \in V$.

Now, set $E^R = \{(v, u) : (u, v) \in E\}$ which are the reversed edges in a graph and define $G^R = (V, E^R)$ as the reversed edge graph. Run $\texttt{Dijkstra}(G^R = (V, E^R), v_0)$ to get a distance ($dist_{bwd}$) map from $v_0 \to u : \forall u \in V$ *in the reversed graph*. This distance map also corresponds to $u \to v_0 : \forall u \in V$ in G.

Hence, for any pair (u, v), the shortest path from $u \to v$ that goes through (u, v) can be given by $dist_{bwd}[u] + dist_{fwd}[v]$.

Justification. This algorithm is correctly as Dijkstra is an algorithm that returns a distance map from a fixed source to every possible destination in the graph from that source. The forward direction is trivially correct as it is a literal application of the algorithm.

Since Dijkstra's algorithm uses a fixed source with variable destinations, then reversing the edges lets us get the shortest paths from variable sources to a fixed destination. This is the exact neccesity for the first half of the path from $u \to v_0$.

Finally, the shortest combined path is also the shortest path from $u \to v_0$ and $v_0 \to v$ when v_0 must be crossed. Hence, computing these two distances seperately and adding them is minimal.

Runtime. This algorithm runs the Dijkstra algorithm twice taking $\mathcal{O}((|V|+|E|) \log |V|)$ runtime and reverses the edges taking $\mathcal{O}(|E|)$ time giving a total runtime of $\mathcal{O}((|V|+|E|) \log |V|)$.