CS2040S Tutorial 9

AY 24/25 Sem 2 — github/omgeta

Q1. (a.) Graph with negative weights.

(b.) Trees (BFS is $O(V + E)$), DAGs (Topo sort is $O(V + E)$)

Q2. (a.) Modify the relax, convering the addition to a multiplication, relax if result is longer rather than shorter

(b.) Set edge weights to $\log f_e$

Q3. (a.) $lp(u) = max_{v \in N(u)}[lp(v)] + 1$

(b.) Post-order DFS is $O(V + E)$

(c.) Toposort and compute $lp(u)$ in reverse toposort order. We can store each $lp(u)$ in an array/hashtable for memoization.

Q4. Two DAGs, one with uphill directed edges, and another with downhill directed edges. Connect nodes from uphill to downhill DAG if an edge originally existed between them.

Q5. (a.) Pre-compute shortest paths to and from all nodes, then find the maximum edge where the sum of edges $\leq D$

(b.) Duplicate the graph twice to force a graph to go through two transient edges, where the transient edges based on binary search for edge lengths.