ECE374 SP23 HW7

Contributors

Zhirong Chen (zhirong4)

Ziyuan Chen (ziyuanc3)

Problem 2

Given a directed acyclic graph (DAG) $G = (V, E)$ with integer (positive or negative) edge weights:

(a) Give an algorithm to find the **shortest** path from a node s to another node t.

(b) Give an algorithm to find the **longest** path from a node s to another node t.

Solution

(a)

We use a variant of Bellman-Ford. Dynamic programming still helps, but the algorithm can be optimized for a DAG.

$\text{DAGSP}(V, E, s, t)$
    $V', E' \leftarrow \text{topoSort}(V, E)$
    $m \leftarrow \text{index}(s, V')$
    $n \leftarrow \text{index}(t, V')$
    for $i \leftarrow m + 1$ **to** n $\quad$ // initialize the memoization array
        $D[i] \leftarrow \infty$
    for $i \leftarrow m$ **to** $n - 1$ $\quad$ // no need to consider nodes before s or after t
        for $j \leftarrow i + 1$ **to** n
            $e \leftarrow (V'[i], V'[j])$ $\quad$ // for each edge between $V'[i]$ and $V'[j]$
            if $e \in E'$
                $D[j] \leftarrow \min(D[j], D[i] + e.\text{weight})$
    return $D[n]$

Runtime analysis. Topological sort takes $O(n + m)$. The nested **for** loop takes $O(n^2)$ time in the worst case. Total time complexity is $O(n^2)$.

(b)

The path with the largest length **"value"** is equivalent to the path with the smallest length value (i.e., the shortest path) on the graph with all edge weights negated. Therefore, a simple adaptation of DAGSP suffices. Time complexity is still $O(n^2)$.

$\text{DAGLP}(V, E, s, t)$
    for $e \in E$
        $e.\text{weight} \leftarrow -e.\text{weight}$
    return $-\text{DAGSP}(V, E, s, t)$
