הבינתחומי הרצליה
IDC HERZLIYA

Data Structures & Algorithms – Problem set 5

Due: 09.01.2022

Problem 1 (20 points):

Let $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ be a weighted directed graph, with no negative cycles, and with weight function $\mathrm{w}: \mathrm{E} \rightarrow \mathbb{R}$. The best source for v is the vertex u such that a path from u to v is the shortest path of all paths that end in v.

Give an $\mathrm{O}(|\mathrm{V}| \cdot |\mathrm{E}|)$-time algorithm, that for each $v \in V$ finds the weight of the shortest path from the best source of u to v.

Note that v might be the best source for itself (with an empty path of weight 0), but since negative weights are allowed, better sources might exist.

Hint: Add a new vertex s to the graph.

Problem 2 (20 points):

Let graph $G = (V, E)$ be a directed graph with no negative cycles. Describe an efficient algorithm to find for a given edge $e_0 = (a, b)$, the weight of the shortest cycle containing it.

Problem 3 (20 points):

Let $G = (V, E)$ be a directed weighted graph with a weight function $w: E \cup \boldsymbol{V} \rightarrow \mathbb{R}^+$ both on the edges **and the vertices**. Note that the weights are non-negative. The weight of a path $P = s, v_1, \dots, v_n, t$ is the sum of the weight of its edges **and its vertices**, not including the first and last vertices, s and t. Given $G = (V, E, w)$, and $s \in V$,suggest an algorithm that computes the shortest path between any pair of vertices s and t in V.

Problem 4 (20 points):

Describe an algorithm based on dynamic programming that solves the following problem, denoted 'heaviest common subsequence'.

Given two strings: $\mathrm{X} = (\mathrm{x}_1, \dots, \mathrm{x}_\mathrm{n})$, and $\mathrm{Y} = (\mathrm{y}_1, \dots, \mathrm{y}_\mathrm{m})$, over the alphabet $\sum$, and a function $\mathrm{w}: \sum \rightarrow \mathrm{R}$ such that $\mathrm{w(a)} > 0$ for every $\mathrm{a} \in \sum$.

Our goal is to find the heaviest common (not-necessarily-continuous) subsequence of X,Y. The weight of a sequence $\mathrm{Z} = (\mathrm{z}_1, \dots, \mathrm{z}_\mathrm{l})$ is $\sum_{\mathrm{i}=1}^{\mathrm{l}} \mathrm{w}(\mathrm{z}_\mathrm{i})$.

Example: for $\Sigma = \{a, b, \dots, z\}$, w gives the ordinal for each letter ($a = 1, b = 2$ etc.), $X = lizard, Y = zebra$, 'za' is a common subsequence, but 'zar' is not because it is not a subsequence of Y. Weight of za is $26 + 1 = 27$ (it is not the heaviest common subsequence).

Problem 5 (20 points):

A robber robs a house. He can carry items of total weight at most T. There are n items in the house with weights $w_1, \dots, w_n$. All numbers are positive integers.

Help the robber snatch as many items as possible. Suggest an algorithm that finds the maximal number of items he can take.

Hint: use a DP table of size n times T.