Homework 5

CS41, Spring 2023 — Due Sunday, February 26

Solve each of the following problems by reducing to a DAG shortest (or longest) path problem. You should be explicit about how you use the given input to algorithmically generate an adjacency list and edge weights and which vertices you are finding a shortest (or longest) path between.

1. The Algoids are trying to rebuild their population after a devastating war with the Chelonians. Currently they are on Planet 0, it is the beginning of year 0, and there are $n-1$ other planets in the system, conveniently numbered $1, \dots, n-1$, and there are known growth rate constants $\alpha_{i,y} > 1$ for all planets $0 \le i < n$ and years $y \in \mathbb{N}$. The Algoids' current population is $p_0 > 0$.

   At the beginning of each year, starting right now, the Algoids have to make a decision: Either remain on the same planet for another year, or move to a new planet. Moving from planet i to planet j will take $t_{i,j} \in [0, 1)$ years, during which the Algoids' population will not grow. Once they reach planet j in year y, their population will grow by a factor of $\alpha_{j,y}^{1-t_{i,j}}$ in the remainder of year y. If they remain on planet j for ℓ more years, then their population will grow by a total factor of $\alpha_{j,y+1} \cdot \cdots \cdot \alpha_{j,y+\ell}$ during those subsequent years.

   Design a $O(n^2 Y)$-time algorithm for determining the Algoids maximum possible population after Y years.

2. The last human city, deep inside the earth, is under attack by flying squid-like machines called Sentinels. The humans have an electromagnetic pulse (EMP) device with which to fight the Sentinels, but it takes time to charge up; a pulse that is released after s seconds will disarm up to $f(s)$ Sentinels, where $f : \mathbb{N} \to \mathbb{N}$ is an increasing function. The value $f(s)$ can be computed in $O(1)$ time.

   The Sentinels come and go in a predictable way, so the humans have recorded an array G telling them that at time t (in seconds), $G[t]$ Sentinels will be present. Thus if the EMP is activated at time t after charging for s seconds (for $s \le t$), then $\min\{f(s), G[t]\}$ Sentinels will be disarmed.

   The goal, given some positive integer time horizon n (which is less than the length of G), is to maximize the total number of Sentinels disarmed in the time range from 0 to n. At time $t = 0$, the EMP is uncharged.

   For example, if $n = 6$, $f(s) = s^3$, and $G = [2, 4, 30, 4, 1, 15, 1]$, then activating the EMP at times 2, 5, and 6 would result in

$$\begin{aligned}
&\min\{2^3, G[2]\} + \min\{(5-2)^3, G[5]\} + \min\{(6-5)^3, G[6]\} \\
&= \min\{8, 30\} + \min\{27, 15\} + \min\{1, 1\} \\
&= 24
\end{aligned}$$

   total Sentinels being disarmed.

   Design a $O(n^2)$-time algorithm that finds the maximum total number of Sentinels that can be disarmed by time $t = n$.

3. Suppose that you have constant-time access to a dictionary of valid English words, i.e., the function

$$\textsc{Dict}(w) = \begin{cases} \text{TRUE} & \text{if } w \text{ is a valid word} \\ \text{FALSE} & \text{otherwise} \end{cases}$$

takes $O(1)$ time. Design a $O(n^2)$-time algorithm to determine whether any given n-letter string s can be decomposed into a sequence of valid English words. For example, your algorithm should return TRUE on input $s =$ `iwastheshadowofthewaxwingslain` and FALSE on input $s =$ `bythfalseazureinthewindowpane`.

We regard strings as simple arrays of characters. In particular, you can access the i^{th} character $s[i]$ in $O(1)$ time, and you can append a character to the end of a string in $O(1)$ amortized time, but retrieving an arbitrary substring $s[i..j]$ would take $O(j - i)$ time.

4. Given two arrays $x[1..m]$ and $y[1..n]$, we wish to find the largest k for which there are indices i and j with $x[i..i + k - 1] = y[j..j + k - 1]$.

For example, if $x = [1, 6, 2, 3, 5, 6, 3]$ and $y = [9, 1, 7, 6, 2, 3, 4, 5, 6, 2]$, then the largest k is 3, corresponding to $x[2..4] = y[4..6] = [6, 2, 3]$.

Describe an $O(mn)$-time algorithm to do this.