Problem Set 8

Please your solutions in the LaTeX template provided. Aim for concise solutions; convoluted and obtuse descriptions might receive low marks, even when they are correct. **There is no coding part to submit.**

Please solve each of the following problems using **dynamic programming**. For each , be sure to define a set of subproblems, relate the subproblems recursively, argue the relation is acyclic, provide base cases, construct a solution from the subproblems, and analyze . Correct but inefficient dynamic programs will be awarded significant partial credit.

For each below, please indicate whether the requested is either:
1 **polynomial**, 2 **pseudopolynomial**, or **exponential** in the size of the .
This categorization will be worth **points per** .

Problem 8-1. [25 points] **Oil Well that Ends Well**

The oil wells of tycoon Ron Jockefeller will produce oil barrels this . Ron has a list of n orders from potential buyers, where the ith a willingness to a_i barrels for a total price of p_i (not per barrel), which may be negative.[1] Each must be filled completely or not at all, and can only be filled once. Ron does not have to sell all of his oil, but he must pay s dollars per unsold barrel in storage costs. Describe an $O(nm)$-time algorithm to determine which orders to fill so that Ron can maximize his profit (which may be negative).

Problem 8-2. [25 points] **Splits Bowling**

In Lecture 15, we introduced **Bowling**: a one-player played on a sequence of n pins, where pin i has integer value ; (possibly negative). The player repeatedly knocks down pins in two ways:

- knock down a single pin, providing ; points; or
- knock down two adjacent pins i and i 1 ,providing ; ; ;41points.

Pins may be knocked down at most once, though the player may choose not to knock down some pins. A Bowling variant, **Split Bowling**, adds a third the player can knock down two pins forming a **split**, specifically:

- knock down two pins i and $j > i +$ 1 if all pins in $\{i$ 1 ,..., 7 − 1 }between them have already been previously knocked down, providing ; ; points.

Describe an $O(n^3)$-time algorithm to determine the maximum possible playing Split Bowling on a given sequence of n pins.

---

[1] Earlier this year, oil futures contract prices went negative: were paying $ to not delivery of oil because demand for oil had fallen dramatically and there was a shortage of places to store oil.

Problem 8-3. [25 points] **Quarter Partition**

Given a set $A = \{a_1, \ldots, a_n\}$ containing n distinct positive integers where $m = \sum_{a_i \in A} a_i$, describe an $O(m^3 n)$-time algorithm to compute a **partition** of A into four subsets $A_1, A_2, A_3, A_4 \subsetneq A$ (where $A_1 \cup A_2 \cup A_3 \cup A_4 = A$) such that the maximum of their individual sums is as small as possible, i.e., such that $\max\left\{\sum_{a_i \in A_j} a_i \mid j \in \{1, 2, 3\}\right\}$ is minimized.

Problem 8-4. [25 points] **Corrupt Chronicles**

Kimmy Jerk is the captain of the USS Exitcost, a starship charged with exploring new worlds. Each day, Capt. Jerk uploads a **captain's log** to the ship's computer: a string of at most m lowercase English letters and spaces, where a **word** in a log is any maximal substring not containing a space.

One day, Capt. Jerk is abducted, and Communications Officer Uhota Nyura goes to the captain's logs looking for evidence. Unfortunately, the log upload system has malfunctioned, and has **corrupted** each of the last n logs by dropping all spaces. Officer Nyura wants to restore the spaces based on Capt. Jerk's past patterns in uncorrupted logs. Given a list L_c of the n corrupted logs, as well as a list L_u of $O(m^2 n)$ uncorrupted logs from before the malfunction, Officer Nyura wants to:

- for each word w appearing in any log in L_u, compute $f(w)$: the positive integer number of times word w appears in L_u, and $f(w)$ is 0 for any word w not appearing in L_u; and
- for each log $\ell_i \in L_c$, compute a **restoration** R_i of ℓ_i (i.e, a sequence of words R_i whose ordered concatenation equals ℓ_i) such that $\sum_{w \in R_i} f(w)$ is maximized over all possible restorations.

Describe an $O(m^3 n)$-time algorithm to restore Capt. Jerk's logs based on the above protocol.

6.006 Introduction to Algorithms

Spring 2020