ADA 2024 Tutorial 4

Diptapriyo, Supratim

This tutorial is more warmup on DPs. What we would like you to do for each of the problems is:-

1. Define the subproblems clearly
2. Write a recursion using the above definition and argue properly about why the recursion is correct (this is the optimal substructure property)
3. Implement using tables and argue runtime.

1 Subset Sum and 3-Partition

For Subset-Sum, see lec6.pdf

3-PARTITION. Given a set of n numbers $A = \{a_1, a_2, \cdots a_n\}$, does there exists a partition of in to three disjoint sets A_1, A_2, A_3 such that

$$\sum_{a_i \in A_1} a_i = \sum_{a_j \in A_2} a_j = \sum_{a_k \in A_3} a_k$$

2 Dictionary

You are given a string of n characters s, which you believe to be a corrupted text document in which all punctuation has vanished (so that it looks something like *"itwasthebestoftimes..."*). You wish to reconstruct the document using a dictionary, which is available in the form of a Boolean function $dict()$: for any string w, $dict(w)$ outputs true if w is a valid word false otherwise. Give a dynamic programming algorithm that determines

whether the string s can be reconstituted as a sequence of valid words. The running time should be at most $\mathcal{O}(n^2)$, assuming each call to $dict()$ takes unit time.

Solution. Let $T[i]$ be 1 if string $s[i \dots n]$ can be reconstituted as a sequence of valid words else 0. Then,

$$T[i] = \max_{j=i \dots n} (dict(i \dots j) = 1 \wedge T[j+1] = 1)$$

The size of the DP table is $1 \times (n+1)$. Filling each entry takes $O(n)$ int the worst case. Therefore time complexity is $O(n^2)$.

3 File Placement Problem

Suppose we want to replicate a file over a collection of n servers,labeled $S_1, S_2, \dots, S_n$. To place a copy of the file at server S_i results in a placement cost of c_i, for an integer $c_i > 0$. Now, if a user requests the file from server S_i, and no copy of the file is present at S_i, then the servers $S_{i+1}, S_{i+2}, S_{i+3}, \cdots S_n$ are searched in order until a copy of the file is finally found, say at server S_j, where $j > i$. This results in an access cost of $j - i$. (Note that the lower-indexed servers $S_{i-1}, S_{i-2}, \dots$ are not consulted in this search.) The access cost is 0 if S_i holds a copy of the file. We will require that a copy of the file be placed at server S_n, so that all such searches will terminate, at the latest, at S_n. We would like to place copies of the files at the servers so as to minimize the sum of placement and access costs. Formally, we say that a configuration is a choice, for each server S_i with $i = 1, 2, \cdots, n-1$, of whether to place a copy of the file at S_i or not. (Recall that a copy is always placed at S_n.)The total cost of a configuration is the sum of all placement costs for servers with a copy of the file, plus the sum of all access costs associated with all n servers. Give a polynomial-time algorithm to find a configuration of minimum total cost.