CS340 Analysis of Algorithms
Fall 2025

Homework:	**6**	**Professor:**	**Elizabeth Dinella**
Due Date:	**11/10/25**	**E-mail:**	`edinella@cs.brynmawr.edu`
Office:	**Park 205**	**URL:**	`https://bmc-cs-340.github.io`

Solutions to all DP problems must clearly define and state the recurrence in the description. In addition, all proofs of correctness should be by induction.

1. Full write-up. Complete a write-up of the chain matrix multiplication algorithm we covered in class. This includes all parts, description, pseudo code, time analysis and correctness proof. The implementation should be bottom-up. In addition to the minimum number of multiplications, you should also provide a way to recover the preferred multiplication order that results in the minumum number of multiplications.

2. For the following questions, give a dynamic programming formulation and brief time analysis only. That is, only a concise description and a DP recurrence are required, accompanied by a short time analysis assuming memoization or bottom-up implementation. Just a brief explanation of the size of the arrays used and what's stored in them is enough.

1. Given a positive integer $n \geq 1$, you may perform the following operations:

   $$n = \begin{cases} n/3 & \text{if n\%3} = 0 \\ n/2 & \text{if n\%2} = 0 \\ n-1 & \text{otherwise} \end{cases}$$

   Find the minimum number of operations required to reduce the number to 1.
2. Given an array of n integers, find the contiguous subarray whose elements give the largest sum. Your algorithm outputs a pair of indices i, j where i is the starting index of the subarray and j the end.
3. You are picking a team to play for your company in the regional soccer tournament. The company organizational chart is a tree. Each employee has a soccer ability rating. Design an algorithm to pick a team (no size limit) with a maximum total ability score subject to the constraint that you do not pick both an employee and his/her immediate boss to minize the possibility of aggression and office drama.

3. Full write-up. Given a string (without spaces) and a dictionary of words, decide if it is possible to segment the string into a space-separated sequence of dictionary words. You may assume a dictionary function `dict(w)` which returns true if `w` is a valid word in $O(1)$. If a segmentation exists, your algorithm should output either the sequence of indices that marks the word breaks, or the actual space-separated string itself. If there is more than one valid segmentation, you may output any one.

Provide an example of running of your algorithm on the following input: “oneinamillion”, dictionary = {“a”,”in”,”mill”,”million”,”lion”, ”on”,”one”}. Illustrate data structure contents during each iteration (that is, show how your DP array is filled).

Please hand in your assignment electronically on Gradescope.