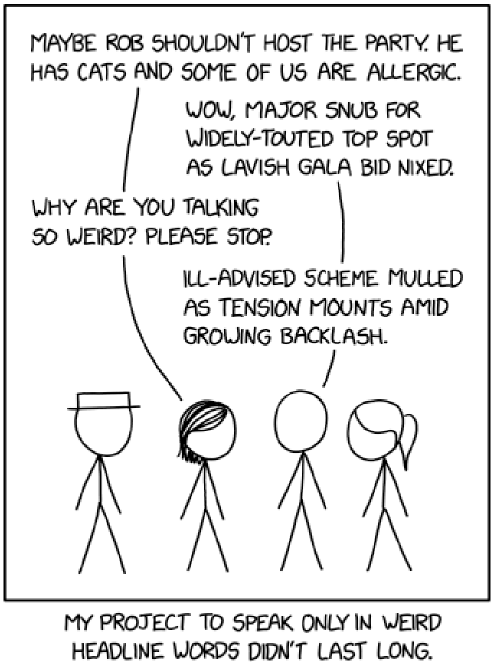
CS140 - Assignment 7

Due: Sunday, 3/26 at 8pm


MAYBE ROB SHOULDN'T HOST THE PARTY. HE HAS CATS AND SOME OF US ARE ALLERGIC.
WOW, MAJOR SNUB FOR WIDELY-TOUTED TOP SPOT AS LAVISH GALA BID NIXED.
WHY ARE YOU TALKING SO WEIRD? PLEASE STOP.
ILL-ADVISED SCHEME MULLED AS TENSION MOUNTS AMID GROWING BACKLASH.
MY PROJECT TO SPEAK ONLY IN WEIRD HEADLINE WORDS DIDN'T LAST LONG.

https://xkcd.com/2584/

For this assignment, you may (and are encouraged to) work with a partner.

For the dynamic programming solutions, in addition to the algorithm, make sure to explicitly state:

- What the table looks like (size and range of values).
- How you initialize the table, i.e., any starting values
- How you fill the table in, i.e., what indices you start at and how you proceed.
- Where the final answer is.

This can either be done with pseudocode or in plain language.

1. **[12 points]** You are given a string of characters $S = s_1, s_2, ..., s_n$ where all non-alphabetic characters have been removed (e.g. "thisisasentencewithoutanyspacesorpunctuation") and a function $\textsc{dict}(w, i, j)$, which takes as input a string w and two indices i and j and returns *true* if the string $w_{i...j}$ is a dictionary word and $false$ otherwise.

   (a) **[10 points]** Give a dynamic programming solution that determines whether the string S consists of a sequence of valid dictionary words. Make sure to explicitly state the size of the table, which elements you fill in first, how you fill in the table, and where the solution is found.

   (b) **[2 points]** State the running time of your algorithm assuming calls to DICT are $O(1)$.

Greedy or DP

One of the problems below can be solved more efficiently using a greedy approach and the other cannot (i.e. you must use dynamic programming). For each problem clearly describe your algorithm and state the run-time. For the dynamic programming problem, make sure to describe the table and how it is filled in. For the greedy problem, argue (a formal proof isn't necessary) that your solution is optimal.

2. **[10 points]** You're going on a road trip with friends. Unfortunately, your headlights are broken, so you can only drive in the daytime. Therefore, on any given day you can drive no more than d miles. You have a map with n different hotels and the distances from your start point to each hotel $x_1 < x_2 < ... < x_n$. Your final destination is the last hotel. Describe an algorithm that determines which hotels you should stay in if you want to minimize the number of days it takes you to get to your destination.

3. **[10 points]** Same setup as above, however, you also want to do some sightseeing along the way. To make sure you don't spend too little or too much time in any one place, you decide to add a penalty for having too much free time. If you travel x miles in a day, then the penalty for that day is $(d - x)^2$. Describe an algorithm that determines the hotel sequence that minimizes the total penalty, that is the sum of the daily penalties over all travel days.

DP on Trees

4. **[15 Points]** You've been asked to design an algorithm for deciding who to invite to a company party. The structure of the company can be described by a tree as follows: the CEO is at the root, below the root are VPs, below them are directors, below them are manages, etc., etc., until you get down to the leaves (summer interns). The tree is not necessarily binary; some non-leaf nodes may have one "child", others two, and others even more.

   To make the party fun, we won't invite an employee along with their immediate supervisor (their parent in the tree). In addition, each person has been assigned a positive real number called their *coefficient of fun*. The goal is to invite employees so as to maximize the total sum of the coefficients of fun of all invited guests, while not inviting an employee with their immediate supervisor.

(a) [**4 points**] Describe a recursive algorithm for this problem (i.e. non-dynamic programming). Assume that the tree is represented as a collection of nodes with links from parents to children and also from children to parents. The tree is passed to you by giving you a reference to the root.

(b) [**8 points**] Describe a DP algorithm for this problem. You may assume that each of the n nodes in the tree has a unique number between 1 and n associated with it. You may also assume that you have a function that will give you a list (array or linked list) of all of the leaves in the tree in time $O(n)$.

(c) [**3 points**] State the running time of your approach with respect to n the number of employees and k the maximum number of children any node has.