CS 473 ✧ Spring 2016

Homework 1

Due Tuesday, February 2, 2016, at **8pm**

- Starting with this homework, groups of up to three students may submit joint solutions. **Group solutions must represent an honest collaborative effort by all members of the group.** Please see the academic integrity policies for more information.
- You are responsible for forming your own groups. Groups can change from homework to homework, or even from (numbered) problem to problem.
- Please make sure the names and NetIDs of *all* group members appear prominently at the top of the first page of each submission.
- Please only upload one submission per group for each problem. In the Online Text box on the problem submission page, you must type in the NetIDs of ***all*** group members, including the person submitting. See the Homework Policies for examples. Failure to enter all group NetIDs will delay (if not prevent) giving all group members the grades they deserve.

- For dynamic programming problems, a full-credit solution must include the following:
  - A clear English specification of the underlying recursive function. (For example: "Let $Edit(i, j)$ denote the edit distance between $A[1..i]$ and $B[1..j]$.") Omitting the English description is a Deadly Sin, which will result in an automatic zero.
  - ***One*** of the following:
    * A correct recursive function or algorithm that computes the specified function, a clear description of the memoization structure, and a clear description of the iterative evaluation order.
    * Pseudocode for the final iterative dynamic programming algorithm.
  - The running time.
- For problems that ask for an algorithm that computes an optimal *structure*—such as a subset, subsequence, partition, coloring, tree, or path—an algorithm that computes only the *value* or *cost* of the optimal structure is sufficient for full credit, unless the problem says otherwise.
- Official solutions will provide target time bounds for full credit. Correct algorithms that are faster than the official solution will receive extra credit points; correct algorithms that are slower than the official solution will get partial credit. We rarely include these target time bounds in the actual questions, because when we do, more students submit fast but incorrect algorithms (worth 0/10 on exams) instead of correct but slow algorithms (worth 8/10 on exams).

---

1. Let's define a *summary* of two strings A and B to be a concatenation of substrings of the following form:

   - ▲SNA indicates a substring SNA of only the first string A.
   - ♦FOO indicates a common substring FOO of both strings.
   - ▼BAR indicates a substring BAR of only the second string B.

   A summary is *valid* if we can recover the original strings A and B by concatenating the appropriate substrings of the summary in order and discarding the delimiters ▲, ♦, and ▼. Each regular character has length 1, and each delimiter ▲, ♦, or ▼ has some fixed non-negative length Δ. The *length* of a summary is the sum of the lengths of its symbols.

   For example, each of the following strings is a valid summary of the strings KITTEN and KNITTING:

   - ♦K▼N♦ITT▲E▼I♦N▼G has length $9 + 7\Delta$.
   - ♦K▼N♦ITT▲EN▼ING has length $10 + 5\Delta$.
   - ♦K▲ITTEN▼NITTING has length $13 + 3\Delta$.
   - ▲KITTEN▼KNITTING has length $14 + 2\Delta$.

   Describe and analyze an algorithm that computes the length of the shortest summary of two given strings $A[1..m]$ and $B[1..n]$. The delimiter length Δ is also part of the input to your algorithm. For example:

   - Given strings KITTEN and KNITTING and $\Delta = 0$, your algorithm should return 9.
   - Given strings KITTEN and KNITTING and $\Delta = 1$, your algorithm should return 15.
   - Given strings KITTEN and KNITTING and $\Delta = 2$, your algorithm should return 18.

2. Suppose you are given a sequence of positive integers separated by plus (+) and minus (−) signs; for example:

$$1 + 3 - 2 - 5 + 1 - 6 + 7$$

   You can change the value of this expression by adding parentheses in different places. For example:

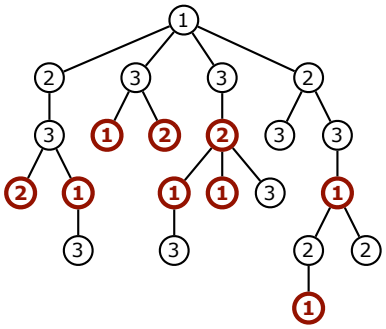
$$1 + 3 - 2 - 5 + 1 - 6 + 7 = -1$$
$$(1 + 3 - (2 - 5)) + (1 - 6) + 7 = 9$$
$$(1 + (3 - 2)) - (5 + 1) - (6 + 7) = -17$$

   Describe and analyze an algorithm to compute the maximum possible value the expression can take by adding parentheses.

   You may only use parentheses to group additions and subtractions; in particular, you are not allowed to create implicit multiplication as in $1 + 3(-2)(-5) + 1 - 6 + 7 = 33$.

3. The president of the Punxsutawney office of Giggle, Inc. has decided to give every employee a present to celebrate Groundhog Day! Each employee must receive one of three gifts: (1) an all-expenses-paid six-week vacation with Bill Murray,[1] (2) an all-the-Punxsutawney-pancakes-you-can-eat breakfast for two at Punxy Phil's Family Restaurant, or (3) a burning paper bag of groundhog poop. Corporate regulations prohibit any employee from receiving exactly the same gift as their direct supervisor. Unfortunately, any employee who receives a better gift than their direct supervisor will almost certainly be fired in a fit of jealousy.

   As Giggle-Punxsutawney's official gift czar, it's *your* job to decide which gift each employee receives. Describe an algorithm to distribute gifts so that the minimum number of people are fired. Yes, you can give the president groundhog poop.



A tree labeling with cost 9. The nine bold nodes have smaller labels than their parents.
The president got a vacation with Bill Murray. This is *not* the optimal labeling for this tree.

   More formally, you are given a rooted tree T, representing the company hierarchy, and you want to label each node in T with an integer 1, 2, or 3, so that every node has a different label from its parent. The *cost* of an labeling is the number of nodes that have smaller labels than their parents. Describe and analyze an algorithm to compute the minimum cost of any labeling of the given tree T.

[1] The details of scheduling n distinct six-week vacations with Bill Murray, all in a single year, are left as an exercise for the reader.