New CS 473: Algorithms, Spring 2015
Homework 3

Due Tuesday, February 17, 2015 at 5pm

---

1. A string w of parentheses **(** and **)** and brackets **[** and **]** is ***balanced*** if it satisfies one of the following conditions:

   - w is the empty string.
   - $w = (x)$ for some balanced string x
   - $w = [x]$ for some balanced string x
   - $w = xy$ for some balanced strings x and y

   For example, the string

   $$w = ([()][]())[()()]()$$

   is balanced, because $w = xy$, where

   $$x = (\;[()]\;[]\;()\;) \qquad \text{and} \qquad y = [\;()\;()\;]\;().$$

   Describe and analyze an algorithm to compute the length of a longest balanced subsequence of a given string of parentheses and brackets. Your input is an array $w[1\,..\,n]$, where $w[i] \in \{(,), [,]\}$ for every index i. (You may prefer to use different symbols instead of parentheses and brackets—for example, $\mathsf{L}, \mathsf{R}, \mathsf{l}, \mathsf{r}$ or $\triangleleft, \triangleright, \blacktriangleleft, \blacktriangleright$—but please tell us what symbols you're using!)

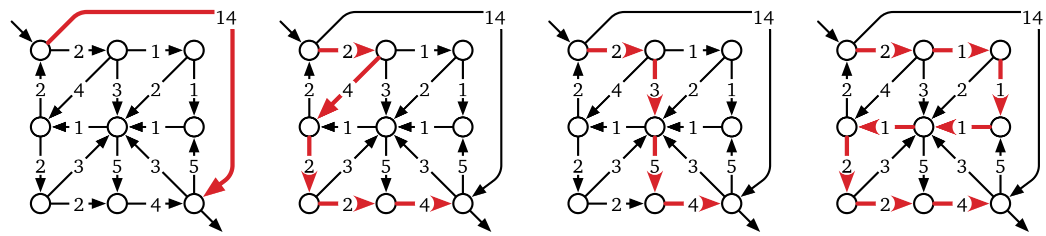
2. Congratulations! You've just been hired at internet giant Yeehaw! as the new party czar. The president of the company has asked you to plan the annual holiday party. Your task is to find exactly k employees to invite, including the president herself. Employees at Yeehaw! are organized into a strict hierarchy—a tree with the president at the root. The all-knowing oracles in Human Resources have determined two numerical values for every employee:

   - $\mathit{With}[i]$ measures much fun employee i would have at the party if their immediate supervisor is also invited.
   - $\mathit{Without}[i]$ measures how much fun employee i would have at the party if their immediate supervisor is *not* also invited.

   These values could be positive, negative, or zero, and $\mathit{With}[i]$ could be greater than, less than, or equal to $\mathit{Without}[i]$.

   Describe an algorithm that finds the set of k employees to invite that maximizes the sum of the k resulting "fun" values. The input to your algorithm is the tree T, the integer k, and the *With* and *Without* values for each employee. Assume that everyone invited to the party actually attends. Do *not* assume that T is a *binary* tree.

3. Although we typically speak of "the" shortest path between two nodes, a single graph could contain several minimum-length paths with the same endpoints.



Four (of many) equal-length shortest paths.

Describe and analyze an algorithm to determine the *number* of shortest paths from a source vertex s to a target vertex t in an arbitrary directed graph G with weighted edges. You may assume that all edge weights are positive and that all necessary arithmetic operations can be performed in $O(1)$ time.

[Hint: Compute shortest path distances from s to every other vertex. Throw away all edges that cannot be part of a shortest path from s to another vertex. What's left?]

New CS 473 Spring 2015 — Homework 3 Problem 1

Name:	NetID:
Name:	NetID:
Name:	NetID:

Describe and analyze an algorithm to compute the length of a longest balanced subsequence of a given string of parentheses and brackets.

New CS 473 Spring 2015 — Homework 3 Problem 2

Name:	NetID:
Name:	NetID:
Name:	NetID:

Describe an algorithm that finds the set of k employees to invite that maximizes the sum of the k resulting "fun" values. The input to your algorithm is the tree T, the integer k, and the *With* and *Without* values for each employee.

New CS 473 Spring 2015 — Homework 3 Problem 3

Name:	NetID:
Name:	NetID:
Name:	NetID:

Describe and analyze an algorithm to compute the *number* of shortest paths from a source vertex s to a target vertex t in a directed graph G with weighted edges.