CS 473: Undergraduate Algorithms, Spring 2009
Homework 2

Written solutions due Tuesday, February 10, 2009 at 11:59:59pm.

- Roughly 1/3 of the students will give oral presentations of their solutions to the TAs. ***Please check Compass to check whether you are supposed give an oral presentation for this homework.*** Please see the course web page for further details.
- Groups of up to three students may submit a common solution. Please clearly write every group member's name and NetID on every page of your submission.
- Please start your solution to each numbered problem on a new sheet of paper. Please *don't* staple solutions for different problems together.
- ***For this homework only:*** These homework problems ask you to describe recursive backtracking algorithms for various problems. ***Don't*** use memoization or dynamic programming to make your algorithms more efficient; you'll get to do that on HW3. ***Don't*** analyze the running times of your algorithms. The ***only*** things you should submit for each problem are (1) a description of your recursive algorithm, and (2) a *brief* justification for its correctness.

---

1. A ***basic arithmetic expression*** is composed of characters from the set $\{1, +, \times\}$ and parentheses. Almost every integer can be represented by more than one basic arithmetic expression. For example, all of the following basic arithmetic expression represent the integer 14:

$$1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1$$
$$((1+1)\times(1+1+1+1+1))+((1+1)\times(1+1))$$
$$(1+1)\times(1+1+1+1+1+1+1)$$
$$(1+1)\times(((1+1+1)\times(1+1))+1)$$

   Describe a recursive algorithm to compute, given an integer n as input, the minimum number of 1's in a basic arithmetic expression whose value is n. The number of parentheses doesn't matter, just the number of 1's. For example, when $n = 14$, your algorithm should return 8, for the final expression above.

2. A sequence $A = \langle a_1, a_2, \ldots, a_n \rangle$ is ***bitonic*** if there is an index i with $1 < i < n$, such that the prefix $\langle a_1, a_2, \ldots, a_i \rangle$ is strictly increasing and the suffix $\langle a_i, a_{i+1}, \ldots, a_n \rangle$ is strictly decreasing. In particular, a bitonic sequence must contain at least three elements.

   Describe a recursive algorithm to compute, given a sequence A, the length of the longest bitonic subsequence of A.

3. A ***palindrome*** is a string that reads the same forwards and backwards, like x, pop, noon, redivider, or amanaplanacatahamayakayamahatacanalpanama. Any string can be broken into sequence of palindromes. For example, the string bubbaseesabanana ('Bubba sees a banana.') can be broken into palindromes in several different ways; for example:

   bub + baseesab + anana
   b + u + bb + a + sees + aba + nan + a
   b + u + bb + a + sees + a + b + anana
   b + u + b + b + a + s + e + e + s + a + b + a + n + a + n + a

   Describe a recursive algorithm to compute the minimum number of palindromes that make up a given input string. For example, given the input string bubbaseesabanana, your algorithm would return the integer 3.