Math 135: Discrete Mathematics, Fall 2010
Homework 5

Due *in class* on Friday, Oct. 29, 2010

---

1. There is a more efficient algorithm (in terms of the number of multiplications and additions) for evaluating polynomials than the one we considered in worksheet 7. It is called **Horner's method**. Consider the following pseudocode for this procedure, which finds a solution to the polynomial $a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$ at $x = c$:

```
procedure Horner(c, a_0, a_1, ..., a_n)
  y = a_n
  for i := 1 to n
      y := y * c + a_{n-i}
  return y
```

   (a) Evaluate $x^4 - 4x^3 + 2x^2 + x + 3$ at $x = 2$ by working through each step of the algorithm and showing the values assigned at least step. (Make sure to write EVERY value a variable gets if it changes during the algorithm.)

   (b) Exactly how many multiplications and additions are used by this algorithm to evaluate a polynomial of degree n at $x = c$? (You don't need to count additions used to increment i in the for loop.)

2. What is the largest n for which one can solve in one second a problem using an algorithm that requires $f(n)$ bit operations, where each bit operation is carried out in 10^{-10} seconds, with these values for $f(n)$?

   (a) $6 \log n$
   (b) $200 \log n$
   (c) n
   (d) $11n$
   (e) n^2
   (f) $18n^2$
   (g) 2^n
   (h) $22 \cdot 2^n$

3. Devise an algorithm that finds all terms of a finite sequence $a_1, \ldots, a_n$ of positive integers that are greater than the sum of all the previous terms of the sequence. Analyze the complexity (number of comparisons, additions, and multiplcations) of your algorithm.

   Note: Algorithms with better complexity/runtime will be given more credit!

4. The traditional Devonian/Cornish drinking song "The Barley Mow" has the following pseudolyrics[1], where *container*$[i]$ is the name of a container that holds 2^i ounces of beer. One version of the song uses the following containers: nipperkin, gill pot, half-pint, pint, quart, pottle, gallon, half-anker, anker, firkin, half-barrel, barrel, hogshead, pipe, well, river, and ocean. (Every container in this list is twice as big as its predecessor, except that a firkin is actually 2.25 ankers, and the last three units are just silly.)

```
BARLEYMOW(n):
  "Here's a health to the barley-mow, my brave boys,"
  "Here's a health to the barley-mow!"

  "We'll drink it out of the jolly brown bowl,"
  "Here's a health to the barley-mow!"
  "Here's a health to the barley-mow, my brave boys,"
  "Here's a health to the barley-mow!"

  for i ← 1 to n
      "We'll drink it out of the container[i], boys,"
      "Here's a health to the barley-mow!"
      for j ← i downto 1
          "The container[j],"
      "And the jolly brown bowl!"
      "Here's a health to the barley-mow!"
      "Here's a health to the barley-mow, my brave boys,"
      "Here's a health to the barley-mow!"
```

(a) Suppose each container name *container*$[i]$ is a single word, and you can sing four words a second. How long would it take you to sing BARLEYMOW(n)? Give the best bound you can in the form $\Theta(f(n))$ for some simple function f.

(b) If you want to sing this song for $n > 20$, you'll have to make up your own container names. To avoid repetition, these names must get progressively longer as n increases.[2] Suppose *container*$[n]$ has $\Theta(\log n)$ syllables, and you can sing six syllables per second. Now how long would it take you to sing BARLEYMOW(n)? Give the best bound you can in the form $\Theta(f(n))$ for some simple function f.

(c) Suppose each time you mention the name of a container, you actually drink the corresponding amount of beer: one ounce for the jolly brown bowl, and 2^i ounces for each *container*$[i]$. Assuming for purposes of this problem that you are at least 21 years old, how many ounces of beer would you drink if you sang BARLEYMOW(n)? Give the best bound you can in the form $\Theta(f(n))$ for some simple function f.

[1] Pseudolyrics are to lyrics as pseudocode is to code.

[2] "We'll drink it out of the hemisemidemiyottapint, boys!"