CS211: Problem Set 6

Due Friday, April 3

I encourage you to discuss these problems with other students. However, the write ups should be your own. Don't write up your solutions while you're collaborating with others. Write them up later—to help ensure that you understand the solution on your own. Cite your collaborators in your submission.

1. **(7 pts) [K&T 6.11]**

2. **(18 pts) [Derived from K&T 6.6] In a word processor, the goal of "pretty-printing" is to take text with a ragged right margin, like this:**

```
Call me Ishmael.
Some years ago,
never mind how long precisely,
having little or no money in my purse,
and nothing particular to interest me on shore,
I thought I would sail about a little
and see the watery part of the world.
```

and turns it into text whose right margin is as "even" as possible, like this:

```
Call me Ishmael.  Some years ago, never
mind how long precisely, having little
or no money in my purse, and nothing
particular to interest me on shore, I
thought I would sail about a little
and see the watery part of the world.
```

To make this precise enough for us to start thinking about how to write a pretty-printer for text, we need to figure out what it means for the right margins to be "even." So suppose our text consists of a sequence of workds $W = w_1, w_2, ..., w_n$ where w_i consists of c_i characters. We have a maximum line length of L. We will assume we have a fixed-width font. A formatting of W consists of a partition of the words in W into lines. In the words assigned to a single line, there should be a space after each word except the last; and so if $w_j, w_{j+1}, ..., w_k$ are assigned to one line, then we should have

$$c_k + \sum_{i=j}^{k-1}(c_i + 1) \leq L$$

We will call an assignment of words to a line $valid$ if it satisfies this inequality. The difference between the left-hand side and the right-hand side will be called the slack of the line—that is, the number of spaces left at the right margin.

(a) **Give an efficient algorithm to find a partition of a set of words W into valid lines, so that the sum of the squares of the slacks of all lines (including the last line) is minimized.**

(b) **Why did we use the sum of the squares instead of just, say, the sum above? That is, what sort of bias does this optimization function create?**

(c) **Write a Python program called pp (not pp.py) that takes two arguments: (1) an integer representing the maximum line length and (2) a file name and print a pretty-printed version using the algorithm. In other words, if `sonnets.txt` is a file of my personal poetry, then**

```
pp 20 sonnets.txt
```

should pretty-print out the poetry to the screen. Note that to make a Python file executable, it needs to have the permissions set to user-executable and have

```
#!/usr/bin/env python
```

as the first line in the Python file.

See the course web page for some test files and a template to begin your Python script.

Use /home/courses/cs211/turnin/username to turn in your assignments.