CS 584/684 Spring 2017 Homework 6 – due noon, Wednesday, May 17 2017

Your solutions to problems 1-4 should be type-set in LaTeX and submitted in both `.tex` and `.pdf` form, with file names `hw6.tex` and `hw6.pdf`. These two files, plus any additional source files invoked from your `.tex` file (such as pictures), should be bundled together into a single `.zip` file named *your-last-name*`-tex-hw6.zip`. Your code for problem 5 should be submitted in a single, separate `.zip` file named *your-last-name*`-code-hw6.zip` with contents as described below in the description of problem 5.

Submit by emailing to `hamialex@pdx.edu` including the zip file as a separate attachment and including "`CS584 HW6`" in the subject line.

All algorithms must be accompanied by proofs of correctness and of running time.

1. Do CLRS exercises 15.3-3 and 15.3-5. Short answers should suffice.

2. Do CLRS problem 15-2. You should achieve a $\Theta(n^2)$ algorithm.

3. Do CLRS problem 15-12.

4. Do CLRS exercise 16.1-5.

5. Implement your solution to CLRS problem 15-2 (longest palindrome subsequence).

Your program should take one command line argument, which is the name of an input file. The format of that file will be as follows:

- First line contains a number C of test cases in the file, where $0 \leq C \leq 10^6$.
- Then come C test cases, each consisting of a single line containing a string of length l containing lower-case alphabetic characters (`a`-`z`), where $1 \leq l \leq 10^3$.

Your program should output (to `stdout`) one line for each test case, of the form "`Case i:` s_i" where s_i is the rightmost longest palindromic subsequence of input case i.

Example Input:

```
3
amanaplanacanalpanama
character
abc
```

Corresponding Example Output:

```
Case 1: amanaplanacanalpanama
Case 2: carac
Case 3: c
```

Warning: If your output format is not correct (even spacing), you will get no credit; this problem will be graded by doing a `diff` against a standard output file.

Place your code (one or more source files) together with a `Makefile` in a fresh subdirectory with the name *your-last-name*`-code-hw6`. Then create a single `zip` archive with the name *your-last-name*`-code-hw6.zip` containing just that directory and its contents. It should be possible to build an executable file called `hw6` and test it on an input file `/path/to/foo` by the following steps:

1. `unzip` *your-last-name*`-code-hw6.zip`

2. `cd` *your-last-name*`-code-hw6`

3. `make`

4. `./hw6 /path/to/foo`