

CSE250, Fall 2011 Assignment 3 Due Wed. 9/21 & Fri. 9/23

This assignment has **both** a hardcopy part and an online-submission part. The hardcopy part is due in class on **Wednesday 9/21**, *not Friday* like the standard for online parts. The online part is due on **Fri. Sep. 23** *by on-line submission* at the (UB CSE-traditional?) **midnight time**.

Office Hours from Week 3 onward, all in Bell 232 until November:

Regan, Tue. 1–3pm and Wed. 2–4pm.

Stojkovic, Tue. 9–11am.

Yao, Thu. 9–11am

Reading. For next week, follow the “Week 3 and beyond” part of the reading directions given with Assignment 1. That is, read the supplementary notes and links on O, o, Θ, Ω notation now on the course webpage (the first is the same as the 1-sheet handout given Wed. 9/14), and make the transtion from Section 2.6 of the text to the more-extensive treatment on my lecture slides, including L’Hôpital’s Rule. Then read the first two sections of Chapter 3.

(1) Text, exercise 1 on pages 178–179. Note that the text is saying “ $O(n)$ ” when it really means $\Theta(n)$ —thus you cannot get away with the trivial (though true!) answer that all four times are $O(n^2)$! ($2 + 2 + 2 + 3 = 9$ pts.)

(2) Suppose $T(n) = n^3 + 5n^2 + 20n - 10$. Note that this differs from the text in exercise 2 on page 179 in having $+5n^2$ not $-5n^2$. To show formally that $T(n) = O(n^3)$, we have to exhibit constants $c > 0$ and $n_0 \geq 0$ such that for all $n \geq n_0$, $T(n) \leq cn^3$.

First consider $c = 2$, and find the minimum value of n_0 that works. Then try $c = 1.1$. How big is the minimum value of n_0 now? ($6 + 6 = 12$ pts.)

(3) Text, exercise 3 on p179, *but modified as follows*: Make one table with $n = 2,048$ and $n = 4,096$, and the second with $n = 4,096$ and $n = 8,192$. Use base 2 for the logarithms, so that all the answers will come out integers. Finally, substitute the constant 3 for the “ O ”, not 1 as the text does. Answer: does this change of constant matter in the final “ $f(2n)/f(n)$ ” column? ($9 + 9 + 3 = 21$ pts., for 42 pts. on this part)

Programming Assignment—due online Fri. 9/23, 11:59pm

Code in C++ a program that reads strings in from a text file named `input.txt` for now (later we will show how to use command-line arguments to pass in any text file). The strings to be read are separated by one-or-more whitespace characters, so that the `>>` operator will read them one-at-a-time. For each string s read, divide s into two pieces s_1 and s_2 , and put them on two lists L_1 and L_2 respectively. Then run through the two lists, joining together pairs of strings into one string, and write the output both to the screen and to an output file, `output.txt`.

For example, if the input file has Ronald Reagan’s declaration **Tear down this wall!** and you break strings in half, then L_1 may have **Te do th wa** and L_2 will have **ar wn is**

ll!, and your output will be `Tedo thwa arwn isll!`. If you broke `wall!` in half the other way, you'd get `Tedo thwal arwn isl!` Whether famous statements are still recognizable after this surgery is the humorous point of the exercise.

Submit your program in a single program `shuffleNNN.cpp` where `NNN` are your initials. Note that your `StringDeque` class can be used to manage the two lists, and you are welcome to re-use your Assignment 2 code here verbatim, putting it in the same file above your `main` exactly as in Assignment 2. If you do so, by rules of ADT's it is not kosher for the class to make visible to the `main` program special index that represents the midpoint of the deque where you started the lists. Instead your `main` should choose a special string s_0 —one you can assume is not going to be read like `QQQQ` or the empty string—to use as the divider. This in turn may make you wish to extend your class by adding methods `peekFront` and `peekRear` that return the front and rear string without popping it.

Following policy from previous terms, now you are welcome to develop your C++ code on your home system, but *you must test it on timberlake so that it compiles and runs there*. Submit it with the syntax

```
submit_cse250 shuffleNNN.cpp
```

except again the `NNN` is not literal. This part is 48 pts., for 90 total on the assignment. Putting the string deque class into separate files `StringDequeNNN.{h,cpp}` is not asked for yet, though it can be 12 pts. extra credit now, 18 with a makefile. Submitting input and output files is optional; if you submit multiple files that's where the ability to submit more than one file on the command line, meant to be hinted with `fooze` last time, comes in.