Homework 01 (Due: Tuesday, February $11, 2020, 11 : 59 : 00$PM)

CSCE 310

Instructions

This assignment consists of 5 analytical problems and 2 programming problems. Your solutions to the analytical problems must be submitted, as one PDF, via webhandin. While handwritten (then scanned) solutions to the analytical problems are acceptable, you are strongly encouraged to typeset your solutions in LaTeX or a word processor with an equation editor. The legibility of your solutions is of great importance. **It is required that your PDF's filename not include spaces, percent signs, parentheses or pound symbols.**

Programming Assignment

Your methods will be tested on the `cse.unl.edu` server, using `gcc version 4.8.5 (SUSE Linux))`. To ensure proper execution, you should test your submission in the webgrader

You will submit `csce310homework01part01.h`, `csce310homework01part02.h`, `csce310homework01part01.cpp`, and `csce310homework01part02.cpp`, along with your PDF, via webhandin. Starter code can be found in Canvas, see the announcement.

`productOfDigits`

`productOfDigits` is a function that should take one `unsigned long long int` as input and return a `unsigned long long int` value. `productOfDigits` should return the product of the digits in the input decimal integer.

`printPermutations`

`printPermutations` is a function that should take one string as input and print out the unique permutations of that string using the Johnson-Trotter algorithm. Each permutation should appear on a separate line. Section 4.3 in *The Design and Analysis of Algorithms* may be of some use. You can assume that the string is already in lexicographic order.

General Guidelines

Sample header, source, and testing files have been provided. You may modify the `.h` and `.cpp` files as needed, but you will only be turning in the four files mentioned above. The webgrader will be compiling the code for each part separately with the command

```
g++ -o /path/to/executable.out /path/to/source/files/*.cpp.
```

If testing locally, know that `test01.cpp` takes a filename as a command-line argument and reads in a number from that file. `test02.cpp` takes one argument: the string.

Written Assignment

Question 1 (10 points)

Adapted from Problem 2.1.7 *in The Design and Analysis of Algorithms*

Gaussian elimination, the classic algorithm for solving systems of n linear equations in n unknowns, requires about $\frac{1}{3}n^3$ multiplications, which is the algorithm's basic operation.

a. How much longer should you expect Gaussian elimination to work on a system of 30 equations versus a system of 10 equations?

b. You are considering buying a computer that is 1000 times faster than the one you currently have. By what factor will the faster computer increase the sizes of systems solvable in the same amount of time as on the old computer?

Question 2 (10 points)

Question 2.1.9 in *The Design and Analysis of Algorithms*

Question 3 (10 points)

Adapted from Problem 2.4.1 *in The Design and Analysis of Algorithms*

Solve the following recurrence relations.

a. $x(n) = x(n-1) + 1$ for $n > 1$ and $x(1) = 9$.

b. $x(n) = 6x(n-1) + 2$ for $n > 0$ and $x(0) = 6$.

c. $x(n) = x(n-1) + n^3$ for $n > 0$ and $x(0) = 6$.

d. $x(n) = x\left(\frac{n}{7}\right) + n$ for $n > 1$ and $x(1) = 7$. Solve for $n = 7^k$.

e. $x(n) = x\left(\frac{n}{9}\right) + 5$ for $n > 1$ and $x(1) = 4$. Solve for $n = 9^k$.

Question 4 (10 points)

Question 3.1.5 in *The Design and Analysis of Algorithms*

Question 5 (10 points)

Question 4.1.5 in *The Design and Analysis of Algorithms*

Question 6 (10 points)

Extra Credit or Honors Contract Question 3.3.5 in *The Design and Analysis of Algorithms*

webgrader Notes

The webgrader should take roughly 60 seconds to complete running.

Your submission's success against each test case will be determined by use of `diff`. Your output must be exact. If you are intending to use debugging output, send that output to stderr, not stdout. Only send output you intend to be matched against the solution output to stdout.

External Resources

cplusplus.com
cppreference.com
Wikibook

Point Allocation

Question	Points
Question 1	10
Question 2	10
Question 3	10
Question 4	10
Question 5	10
`productOfDigits`	
Test Cases	1 × 15
Compilation	10
`productOfDigits` Total	25
`printPermutations`	
Test Cases	1 × 15
Compilation	10
`printPermutations` Total	25
Total	100