Homework 03 (Due: Wednesday, March 18, 2020)

CSCE 310

Instructions

This assignment consists of 4 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, pound symbols, or parentheses.**

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 `csce310homework03part01.h`, `csce310homework03part02.h`, `csce310homework03part01.cpp`, `csce310homework03part02.cpp` (and maybe `csce310homework03part03.h` and `csce310homework03part03.cpp`), along with your PDF, via webhandin. Starter code can be found in Canvas, see the announcement.

`maximumST`

`maximumST` is an adaptation of Exercise 9.3.5 on page 332. `maximumST` is a function that should take an $n \times n$ vector (representing the adjacency matrix of a graph) as input and return the value of the maximum spanning tree of the graph defined by the $n \times n$ vector.

`allPairsSP`

`allPairsSP` is a function that should take an $n \times n$ vector (representing the adjacency matrix of a graph) and a value i between 1 and n as input and return the $n \times n$ vector representing the matrix computed after the i^{th} stage of Floyd's algorithm. For the purposes of this program, infinity (∞) will be represented as -1.

`hornersRule (15 Points Extra Credit, or Honors Contract)`

`hornersRule` is a function that should take a vector of integers (coefficients of a polynomial) and another integer as input and return the value of polynomial, evaluated at that particular integer.

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/six files mentioned above. The webgrader will be compiling the code with the command `g++ -o /path/to/executable.out /path/to/source/files/*.cpp` for each part, but I will only be copying the files I asked for out of your submission and into separate directories for Part 1, Part 2, and Part 3.

Written Assignment

Question 1 (10 points)
Question 6.3.5 in *The Design and Analysis of Algorithms*

Question 2 (10 points)

(a) Construct a $2-3$ tree for the list `a,m,b,i,d,e,x,t,r,o,u,s,l,y`. Use the alphabetical order of the letters and insert them successively starting with the empty tree.

(b) Assuming that the probabilities of searching for each of the keys (i.e., the letters) are the same, find the largest number and the average number of key comparisons for successful searches in this tree.

Question 3 (10 points)
Question 6.3.9 in *The Design and Analysis of Algorithms*

Question 4 (10 points)
Question 9.1.12 in *The Design and Analysis of Algorithms*

Point Allocation

Question	Points
Question 1	10
Question 2	10
Question 3	10
Question 4	10
`maximumST`	
Test Cases	1×25
Compilation	5
`maximumST` Total	30
`allPairsSP`	
Test Cases	1×25
Compilation	5
`allPairsSP` Total	30
Total	100