

CptS 223 - Advanced Data Structures in C++

Written Homework Assignment 1: Math Review, Big-O, Recursion and General Linux/Git Topics

Assigned: Monday, February 1, 2021

Due: Sunday, February 14, 2021

I. Problem Set:

1. (15, -1 pts/rank) Order the following set of functions by their growth rate (from fastest to slowest - rank 1 - 12, where 1 is the fastest and 12 is the slowest). Hint: you can plot their curves in a X-Y axis using <http://fooplot.com/>:

Unordered Complexities	<u>Ordered</u> Complexities
N	5
\sqrt{N}	2
$N^{1.5}$	8
N^2	9
$N \log N$	6
$N \log(\log(N))$	4
$N \log^2 N$	7
$2/N$	3
2^N	12
$2^{(N/2)}$	11
37	1
$N^2 \log(N)$	10

2. (15 pts) A program takes 35 seconds for input size 20 (i.e., $n=20$). Ignoring the effect of constants, approximately how much time can the same program be expected to take if the input size is increased to 100 given the following run-time complexities?
 - a. $O(N)$ = 175s

$$b. O(N + \log N) = 176.54s$$

$$c. O(N^3) = 4375s$$

$$d. O(2^N)^1 = 4.2312404E25s$$

3. (10 pts) How many nodes in a complete trinary tree of depth 5? Hint: use geometric series.

$$\text{sum}(3^{(d-1)}) \text{ from } 1 \text{ to } d: 3^0 + 3^1 + 3^2 + 3^3 + 3^4 = 121$$

4. (15 pts) Write a simple recursive function to calculate (and return) the height of a general binary tree T. The height of a tree T is defined as the number of levels below the root. In other words, it is equal to the length of the longest path from the root (i.e., number of edges along the path from the root to the deepest leaf). Note that the term “nodes” is used to include both internal nodes and leaf nodes. You can assume the following tree node structure:

```
class Node
{
    Node *left; // points to the left subtree
    Node *right; // points to the right subtree
}
```

Your answer can be in C++ syntax or in the form of a generic pseudocode.

```
int height(Node n) {
    if (n == NULL) return 0;
    return maximum(height(n->left), height(n->right)) + 1;
}
```

5. (15 pts) Rewrite the pseudocode presented in class for the Fibonacci numbers *without* recursion (hint: use loop) and discuss the pros and cons of recursion compared to iteration.

```
Int fibonacciIteration(int n){
    Int f1 = 0;
    Int f2 = 1;
    Int fibonacci;
    For(int i = 0; i < n; i++) {
        fibonacci = f1 + f2;
        f1 = f2;
        f2 = fibonacci;
    }
    Return fibonacci;
}
```

6. (10 pts) What is Git and what is the purpose of using Git in general?

Git is a server-based tool that allows programs to be stored in a way that can be accessed by many different people and devices. It allows the ability to clone a program and work on the cloned portion, then merge it back with the main program adding the new changes. It is used so that multiple people can simultaneously work on the same program without interfering with each other, and so that projects can be accessed from any device with less hassle than moving it from device to device as needed

7. (10 pts) What is the Linux tool gdb? What is the difference between cmake and make?

Gdb is the GNU debugger, and is a tool used to debug code in Linux. Make is a tool that builds software, but it is platform dependent. CMake is compiler and platform independent, but it doesn't build the software, it generates the proper builder.

8. (10 pts) How do `argc` and `argv` variables get set if the program is called from the terminal and what values do they get set with?

```
int main(int argc, char* argv[])
{
    return(0);
}
```

When you call the program from the terminal, you can add arguments after the call separated by spaces. `argc` will then be the number of arguments in the call, including the program call itself. `argv` is an array of strings of the arguments, `argv[0]` being the program call `"/programName"`. If no arguments are added to the call then `argc = 1` and `argv` just has the program call as its only string.

II. Submitting Written Homework Assignments:

1. On your local file system, create a new directory called HW1. Move your HW1.pdf file in to the directory. In your local Git repo, create a new branch called HW1. Add your HW1 directory to the branch, commit, and push to the remote origin which is your private GitHub repo.
2. Do not push new commits to the branch after you submit your link to Canvas otherwise it might be considered as late submission.
3. Submission: You must submit a URL link to the branch of your private GitHub repository. Please add the GitHub accounts of the instructor and two TAs (see Syllabus) as the collaborators of your repository. Otherwise, we won't be able to see your repository.

III. Grading Guidelines:

This assignment is worth 100 points. We will grade according to the following criteria:

- See above problems for individual point totals.