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	Ordered Complexities
N	9
√N	10
N^1.5	5
N^2	4
N log N	7
N log(log(N))	8
N log^2 N	6
2/N	12
2^N	1
2^(N/2)	2
37	11
N^2 log(N)	3

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 runtime complexities?

a. O(N)

Answer: 175 seconds

b. O(N + log N)

Answer: 167.59753 seconds

c. O(N^3)

Answer: 4375 seconds

d. O(2^N)[1]

Answer: 4.2312404*10^25 seconds

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

= (3^0)+(3^1)+(3^2)+(3^3)+(3^4)+(3^5) nodes = 364 nodes

Answer: 364 Nodes

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.

```
#include<iostream>
using namespace std;

int findHeight(Node* rootNode){
        //base case
        if(rootNode==NULL){
                return 0;
        }
        else{
```

[1] You might need an online calculator with arbitrarily large numbers for this one. Scientific notation and 8 significant figures is just fine.

```
                int a = findHeight(rootNode->left);
                int b = findHeight(rootNode->right);
                return (max(a,b)+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.

```
Fibonacci(n)  //note, this is pseudocode
        if(n<=1)
                return 1
        else
                int a = 0, b = 1, count = 2, c = 0
                while(count<=n)
                        if count =2
                                then c = b+1
                        else
                                 c = b +a
                        update a and b to a = b and b = c
                        count++
                end while
                return c
        end else
```

The pros of recursion compared to iteration is that the code size is usually reduced, which can be seen from the difference in length of pseudocode for the Fibonacci function between the recursive pseudocode and iterative one above. Additionally, another pro of recursion is that it is easier to traverse trees and it can reduce time complexity when compared to iteration. Some of the cons of recursion is that due to repeated storing values on the stack it takes up more space and also the use of the stack makes it slower than iteration, with a risk of Stack Overflow Error.

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

   Git is a version-control tool to find code changes in files by programmers. The purpose of Git is for programmers to upload files and modify them across devices and systems, and this also allows for multiple collaborators to work on projects together.

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

The Linux tool gbd is the GNU debugger which allows you to debug and follow the program during execution and add breakpoints into the code. The difference between cmake and make is that make is a singular platform build system but cmake stands for Cross-platform make, and cmake generates build systems through a compiler-independent and platform-independent method.

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);
}
```

The argc and argv variables get set with the text entered into the command line, with the argv array storing each ‘word’ and the argc integer storing the number of elements in argv, or the number of strings. argc is always at least one as argv[0] is always the name of the file or filepath.

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.