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	2/N
√N	37
N^1.5	√N
N^2	N
N log N	N log(log(N))
N log(log(N))	N log N
N log^2 N	N log^2 N
2/N	N^1.5
2^N	N^2
2^(N/2)	N^2 log(N)
37	2^(N/2)
N^2 log(N)	2^N

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)

b. O(N + log N)

c. O(N^3)

d. O(2^N)[1]

Answers –

a. 175 seconds
b. 175 seconds
c. 4375 seconds
d. $4.231240368 * 10^{25}$ seconds

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

Answer – Total nodes = $3^0 + 3^1 + 3^2 + 3^3 + 3^4 + 3^5 - 1 = 364$

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 Node :: calculate_height(Node *ptr)
{
    if(ptr == nullptr)
    {
        return 0;
    }

    else
    {
        int left = calculate_height(ptr->left);
        int right = calculate_height(ptr ->right);
```

```
                if(right > left)
                {
                        return right + 1;
                }
                else
                {
                        return left + 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.

Answer -

```
Fibonacci(n)
int f1 = 0;
int f2 = 1;
int fib;
for( i = 2; i <n ; i++)
{
    fib = f1 + f2;
    f1 = f2;
    f2 = fib;
}
return fib;
```

An iterative function is preferred if we're concerned about the time complexity. The space complexity of recursive programs is higher than iterations. Normally, iteration is faster than recursion.In Recursion the problem is divided into smaller problems until we reach the base case.In recursion, the stack is used to store local variables when the function is called.A recursive program is more readable.

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

Git is a free and open source software. It is used for collaborating on projects. Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

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

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

GDB stands for GNU debugger. It is used to debug programs in C and C++. A debugger is a program that runs other programs, allowing the user to exercise control over these programs, and to examine variables when problems arise. GDB allows you to run the program up to a certain point, then stop and print out the values of certain variables at that point, or step through the program one line at a time and print out the values of each variable after executing each line. GDB uses a simple command line interface.

Cmake is a system to generate make files based on the platform (i.e. CMake is cross platform) which you can then make using the generated makefiles.In make you directly write a Makefile for a specific platform that you are working with.