CS 202 Iditarod Challenge 2

Kelby Hubbard

March 24, 2020

- Repository Link: `https://github.com/krhubbard2/CS202/tree/master/Iditarod2`
- Git Commits: `https://github.com/krhubbard2/CS202/commits`
- This homework took approximately 2 hours to complete.

1 Design

Overall design for this program was pretty simple. I first wrote the Ackermann's function to the best of my ability and then just cout the result. I in a sense guessed and checked until I was sure the function was correct.

2 Post Mortem

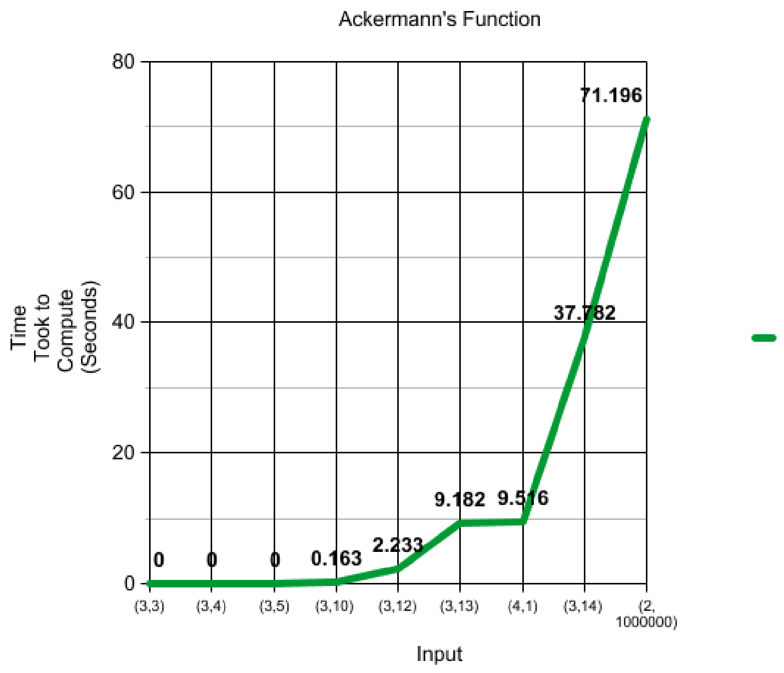
This program was very nice to write. I implemented my stopwatch class from previous assignments to time the calculations and a lot of the time spent on this program was just inputting numbers into the function and timing how long the computer takes to solve it, if it can. I often got a Core Dumped error if the calculation was too large.

3 Recursion Problems

3.1 Sample Output

Listing 1: Sample Program Output

```
200003
Finished job: Tue Mar 24 14:18:11 2020
Elapsed time: 71.1964s
```



3.2 Git Commit Messages

Date	Message
2020-03-24	Write ack(m,n)
2020-03-24	Add stopwatch
2020-03-24	Test largest numbers machine can handle
2020-03-24	Created graph

3.3 Source Code

```cpp
// Kelby Hubbard
// CS202
// March 24, 2020
// Iditarod Challenge 2

#include <iostream>
#include "stopwatch.hpp"

using std::cout;
using std::endl;

//Ackermann's Function
int ack(int m, int n)
{
  if (m == 0)
  {
    return n + 1;
  }
  else if ((m > 0) && (n == 0))
  {
    return ack(m - 1, 1);
  }
  else if ((m > 0) && (n > 0))
  {
    return ack(m - 1, ack(m, n-1));
  }
};

int main()
{
  StopWatch stopwatch;
  stopwatch.starttimer();
  cout << ack(3,13) << endl;
  stopwatch.stoptimer();
  stopwatch.elapsed();


  //Results
  //(3,3) = 61 : Elapsed time: 0.000015971s
  //(3,4) = 125 : Elapsed time: 0.000074735s
  //(3,5) = 253 : Elapsed time: 0.000263427s
  //(3,10) = 8189 : Elapsed time: 0.162673s
  //(3,12) = 32765 : Elapsed time: 2.23392s
  //(3,13) = 65533 : Elapsed time: 9.18236s
  //(4,1)= 65533 : Elapsed time: 9.51603s
  //(3,14) = 131069 : Elapsed time: 37.7826s
  //(2,1000000) = 200003 : Elapsed time: 71.1964s
  //(3,15) : Segmentation fault (core dumped)
  //(4,2) : Segmentation fault (core dumped)


}
```

3.4 Stopwatch Header

```cpp
// Kelby Hubbard
// CS202
// Jan. 26, 2020
// HW001 -- Time It II

#ifndef STOPWATCH_HPP_
#define STOPWATCH_HPP_


#include <chrono>
#include <ctime>
#include <iostream>
using std::cout;
using std::endl;
#include <random>

class StopWatch
{
public:

  std::chrono::system_clock::time_point _start;
  std::chrono::system_clock::time_point _end;

void starttimer();
void stoptimer();
void elapsed();
double mbps();
};






#endif
```

3.5 Stopwatch Source

```cpp
// Kelby Hubbard
// CS202
// Jan. 26, 2020
// HW001 -- Time It II

#ifndef STOPWATCH_HPP_
#define STOPWATCH_HPP_


#include <chrono>
#include <ctime>
#include <iostream>
using std::cout;
using std::endl;
#include <random>

class StopWatch
{
public:

  std::chrono::system_clock::time_point _start;
  std::chrono::system_clock::time_point _end;

void starttimer();
void stoptimer();
void elapsed();
double mbps();
};






#endif
```