

Lab 6

Each lab will begin with a recap of last lab and a brief demonstration by the TAs for the core concepts examined in this lab. As such, this document will not serve to tell you everything the TAs will do in the demo. It is highly encouraged that you ask questions and take notes. In order to get credit for the lab, you need to be checked off by the end of lab. For non-zero labs, you can earn a maximum of 3 points for lab work completed outside of lab time, but you must finish the lab before the next lab. For extenuating circumstances, contact your lab TAs and Instructor.

(4 pts) Iteration vs. Recursion

You will practice writing and timing iterative and recursive versions of the same function. You will use the following code to time both versions of your solution to the Fibonacci problem described below.

```
#include <iostream>
#include <sys/time.h>
#include <cstdlib>

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

int main() {
    typedef struct timeval time;
    time stop, start;

    gettimeofday(&start, NULL);
    //Time your iterative or recursive function here.

    gettimeofday(&stop, NULL);

    if(stop.tv_sec > start.tv_sec)
        cout << "Seconds: " << stop.tv_sec-start.tv_sec << endl;
    else
        cout << "Micro: " << stop.tv_usec-start.tv_usec << endl;

    return 0;
}
```

First, you will write a function called, **fib_iter()**, that has one parameter, n , of type `int`, and returns the n th Fibonacci number, F_n . The Fibonacci numbers start with F_0 is 0, F_1 is 1, F_2 is 1, F_3 is 2, F_4 is 3, F_5 is 5, F_6 is 8, etc...

Formally stated:

$$F_{i+2} = F_i + F_{i+1} \text{ for } i = 2, 3, \dots; \text{ where } F_0 = 0 \text{ and } F_1 = 1$$

Your iterative version of the function should use a loop to compute each Fibonacci number, discarding numbers along the way, until you reach your desired number.

Next, you will write a recursive version of the function called, **fib_recurs()**, that also takes one parameter, *n*, of type int, and returns the *n*th Fibonacci number, F_n .

Using the code supplied above, **time the iterative and recursive solutions for finding the 1st, 5th, 15th, 25th, and 45th Fibonacci numbers.** Determine how long each function takes, and compare and comment on your results.

Show your TA a trace through the recursive solution for $n=5$. Additionally, produce a table of times for $n = 1, 5, 15, 25,$ and 45 for each version of the function.

(6 pts) Fibonacci Application

Design and implement your solution for the following problem.

You're standing at the base of a staircase and would like to get to the top. A small stride will take you up one step while a large stride takes you up two steps. You would like to count the number of possible ways you could climb the entire staircase based on different combinations of large and small strides.

For example, a staircase with three steps can be climbed in three different ways:

- three small strides
- one small stride followed by one large stride
- one large followed by one small stride

How could you apply the Fibonacci number series to this problem? What are the base cases for counting the ways you can climb stairs by going one stair or two stairs at a time? How many different ways can you climb 4 stairs? 5 stairs?