# CS 201, Spring 2017
## Homework Assignment 2

Due: 23:59, April 25, 2017

In this homework, you will study the problem of finding the $n^{th}$ Fibonacci number. Two alternative algorithms that solve this problem are discussed below (also in class). Each algorithm has a different time complexity. The goal of this homework is to simulate the growth rates of both algorithms using different inputs.

**Algorithm 1:** An iterative algorithm which works in O(n) time.

**Algorithm 2**: A recursive algorithm which works in $O(2^n)$ time.

It is possible to improve the run time of the recursive solution using data structures but for the scope of this homework, you are expected to use the version discussed in class.

**ASSIGNMENT:**

1. Study the algorithms and understand how the upper bounds are found for the running time of each solution.
2. Use the implementations given in the slides and write a driver (main) function that calls these functions. Then, run each solution and record the execution times when different *n* values are used. You are expected to try many different input sizes, both small inputs and very large inputs (as large as around 100 000 for the iterative solution and around 55 for the recursive solution), to observe the effects of different growth rates.
3. Use these results to generate a plot of running time (y-axis) versus the input size *n* (x-axis). Specifically, you are expected to produce a plot as in Figure 2.3 of the handout chapter on algorithm analysis.
4. Provide the specifications of the computer you used to obtain these execution times. You can use any computer with any operating system for this assignment.
5. Plot the expected growth rates obtained from the theoretical analysis (as given for each algorithm above) by using the same *n* values that you used in your simulations.
6. Compare the expected growth rates and the obtained results, and discuss your observations in a paragraph.

You can use the following code segment to compute the execution time of a code block. For these operations, you must include the ctime header file.

```
//Store the starting time
double duration;
clock_t startTime = clock();

//Code block
...

//Compute the number of seconds that passed since the starting time
duration = 1000 * double( clock() - startTime ) / CLOCKS_PER_SEC;
cout << "Execution took " << duration << " milliseconds." << endl;
```

An alternative code segment to compute the execution time is as follows. For these operations, you must include the chrono header file.

```
//Declare necessary variables
std::chrono::time_point< std::chrono::system_clock > startTime;
std::chrono::duration< double, milli > elapsedTime;

//Store the starting time
startTime = std::chrono::system_clock::now();

//Code block
...

//Compute the number of seconds that passed since the starting time
elapsedTime = std::chrono::system_clock::now() - startTime;
cout << "Execution took " << elapsedTime.count() << " milliseconds." << endl;
```

**NOTES:**

This assignment is due by 23:59 on April 25th, 2017. This homework will be graded by your TA Can Fahrettin Koyuncu (koyuncu@cs.bilkent.edu.tr).

1. Before the deadline, you should email your homework to Can Fahrettin Koyuncu with subject line CS 201 - HW2.

   No hardcopy submission is needed. The standard rules about late homework submissions apply. Please see the course syllabus for further discussion of the late homework policy as well as academic integrity.

2. In this assignment, you must submit a report (as a pdf file) that contains all information requested above (plots, computer specification, discussion) and a cpp file that contains the main function that you used as the driver in your simulations in a single archive file (you do not need to submit the solution functions).

   Make sure that each file that you submit (each and every file in the archive) contains your name and student number at the top.

3. This homework will be graded by your TA Can Fahrettin Koyuncu (koyuncu@cs.bilkent.edu.tr). Thus, you may ask your homework related questions directly to him.