

## FUNDAMENTALS OF ARTIFICIAL INTELLIGENCE - CS161

## Programming Assignment 1 - Due 11:59 p.m., Tuesday Jan 15

DO NOT CONSULT ANY OUTSIDE REFERENCES OR SOURCES FOR THIS PROBLEM SET

This problem concerns the famous Fibonacci sequence. The first few elements of the sequence are 1 1 2 3 5 8 13 21 34... Each term is the sum of the two previous terms. This sequence is ubiquitous in nature.

1. Write a single pure LISP function, called FIB, that takes a single numeric argument N, and returns the Nth Fibonacci number. For example (FIB 3) should return 2. Test your program on at least the first 20 Fibonacci numbers. Also test your program for larger values of N. What happens? Why?

2. Write a single pure LISP function, called SUMS, that takes a single numeric argument N, and returns the number of additions required by your FIB function to compute the Nth Fibonacci number. Design the recursion for SUMS by examining your FIB code. SUMS should not call any auxiliary functions. Test your program on at least the first 10 values. What is the relationship between the values returned by FIB and SUMS? Explain why this is the case.

3. Write a purely applicative LISP program that computes the Nth Fibonacci number using approximately N additions. In other words, the running time of your program should be proportional to N. Write a top-level function, called FASTFIB, that takes a single numeric argument N, and returns the Nth Fibonacci number. You may define and use only one auxiliary function. You may only use constructs covered in class and recitation. You may not use any global variables or looping constructs. Try to minimize the number of arguments to your functions. My solution uses only two single-argument functions. Note that you are not expected to develop a closed-form analytic expression for the Nth Fibonacci number, but rather to formulate a recursive solution. Test your program for N = 1 through 10, and also 20, 30, 40, 50, 60, 70, 80, 90, and 100. Time the execution of your FIB function from part 1 and estimate how long your FIB function would take to compute the 100th Fibonacci number.

Submit a single file with your commented code, and the answers to the questions asked above, via CCLE. Be sure to name your functions the same as specified above, as they will be tested automatically. Your programs should be written in good style. Provide an overall comment explaining your solutions. Any characters following a semicolon (;) on a line are treated as comments. Multi-line comments, such as the answers to questions, can be delimited by #| and |#. Furthermore, every function should have documentation explaining precisely what its arguments are, and what value it returns as a function of its arguments. In addition, you should use meaningful identifier names. Finally, the physical layout of the code on the page is very important for making LISP programs readable. Make sure that you use blank lines between functions and indent properly. Programming style will be an important consideration in grading the assignments in this class.