CS-430 HW3 Fall 2022 19 points

Submission instructions

- Due date: Friday, Nov. 18, 11:59 pm Central Time (i.e. local time in Chicago)

- *Late submissions and submissions violating these instructions will NOT be accepted.*

- ***Absolutely no handwritten submissions. No credit will be given for such submissions.***

- Teamwork is allowed (max. 4 students/team). Individual submissions are also OK.

- Upload the following files to Blackboard:
  *- (1) your HW report (**pdf format only; the reports in formats other than pdf will be disregarded**);*
  - (2) the source codes of your programs.
  The Beacon students: upload your submissions to LMS.

*- **One submission per team only.** Write down names, A#, and section numbers (i.e. live, online, Beacon) of all the team members on the front page. Do **not** submit multiple copies of your assignment (e.g. by each team member). It is very confusing and will be penalized. **Clearly indicate how each team members contributed to your teamwork.***

- If you use any additional materials to solve the HW problems (e.g. textbooks, research papers, websites, etc.), reference them.

*- **Hao Ding (hding9@hawk.iit.edu)** is responsible for grading this assignment. Feel free to ask questions if you have any doubts but don't send him or me:*
  - Your partial solutions with inquiries "Is that what you expect?".
  - Questions the answers to, may give explicit hints on how to solve the problems.

1. The Fibonacci sequence *0, 1, 1, 2, 3, 5, 8, 13, 21, 34, …* is defined for $n \geq 2$ by the recurrence $F(n)=F(n-1)+F(n-2)$ with $F(0)=0$ and $F(1)=1$.

   (a) (2 points) Calculate recursively $F(n)$ by applying the top-down procedure *Fib( )* which calls itself. Use your favorite programming language to write *Fib( )*. Test *Fib( )* for different numerical values of n to make sure that $Fib(n)$ correctly calculates $F(n)$. You can use the formula for $F(n)$ given in item (f) below to check your numerical results.

   (b) (2 points) Draw the recursion tree for $Fib(5)$. Let's denote by $T(n)$ the total number of calls to *Fib( )* needed to calculate $F(n)$ (note that $T(n)$ includes the original call to $Fib(n)$ at the root). What is the value of $T(5)$?

   (c) (4 points) What is the recurrence for $T(n)$? What are the initial conditions of this recurrence?

   (d) (2 points) Use *Fib( )* to empirically verify that $T(n)=2F(n+1)-1$.

   (e) (4 points) Given the recurrence for $T(n)$ found in item (c) above, prove by induction that the formula $T(n)=2F(n+1)-1$ is correct.

(f) (2 points) Use the formula given in item (d) to find the asymptotic complexity *T(n)=O(?)* of *Fib( )*. The closed-form expression for the Fibonacci numbers $F(n) = \frac{\phi^n - \psi^n}{\sqrt{5}}$, where $\phi = \frac{1+\sqrt{5}}{2}$ and $\psi = \frac{1-\sqrt{5}}{2}$, may be helpful.

(g) (2 points) Implement in your favorite programming language the bottom-up iterative procedure *Better_Fib( )* with "memoization" for better performance. Test *Better_Fib( )* for different numerical values of n to make sure that it correctly calculates *F(n)*. You can use the formula for *F(n)* given in item (f) above to check your numerical results.

(h) (1 point) What is the asymptotic complexity $T_{BF}(n)=O(?)$ of *Better_Fib( )?*

Submit source codes of all programs you have used to produce your results.