# CS:3330 Homework 2, Spring 2018

## Due in class on Thu, Feb 1

---

1. Prove by induction: for all integers $n \geq 0$,
$$\begin{pmatrix} F_n \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n \cdot \begin{pmatrix} F_0 \\ F_1 \end{pmatrix}.$$
Recall that this equation was very helpful in the design of an efficient algorithm to solve the ModuloFibonacci problem.

2. Solve Problem 1.32(a) from the textbook.
**Notes:** (a) You cannot assume that there is an efficient implementation of the square root function that you can use. (b) Think about what the input size is and that will help you determine roughly what your target running time ought to be. (c) Think binary search.

3. Analyze the following code fragment and write down the running time of this code fragment using the $\Theta$ notation, as a function of $n$. Please show your work in order to receive partial credit.
**Note:** Depending on the approach you take, it is possible that you will find this *Stirling's Approximation* useful:
$$\ln(n!) = n \ln n - n + O(\ln n).$$

```
for k ← 1 to n do
    for i ← 1 to n do
        j ← 1
        while j ≤ i do
            print("hello")
            j ← 2 × j
```

4. This problem has nothing to do with the material covered in class or textbook. But, it is an algorithm-design problem that is asked sometimes in job interviews. I will occasionally assign problems of this type for homework to get you to practice "algorithmic thinking" in general.

   You are given a sorted list $L$ of numbers (say, sorted in increasing order). The problem is to find two indices $i$ and $j$, $i \neq j$ such that $L[i] + L[j] = 0$. Of course, it is possible that no such $i$ and $j$ exists (e.g., if all elements in $L$ are positive). In that case, the output should indicate that no such $i$ and $j$ exist.

   Your task is to describe an $O(n)$ time algorithm for this problem and then present a brief argument as to why this algorithm runs in $O(n)$time.

---