Name:

Student ID #:

- You are expected to work on this assignment on your own
- Use pseudocode, Python-like or English to describe your algorithms. Absolutely no C++/C/Java
- When designing an algorithm, you are allowed to use any algorithm or data structure we explained in class, without giving its details, unless the question specifically requires that you give such details
- Always remember to analyze the time complexity of your algorithms
- Homework has to be submitted electronically on Gradescope by the deadline. No late assignments will be accepted
Problem 1. (0 points) Go to Gradescope (https://gradescope.com/ or follow the link from the class CS 218 webpage) and sign up using the entry code ME7YE5. Please make sure you type your full name and student ID correctly in Gradescope. Once you have completed this homework, submit the PDF via Gradescope by October 10th, 2018, 11:59pm
Problem 2. (10 points) Give a tight bound (using the big-theta notation) on the number of Hi’s produced by the following method as a function of $n$. For simplicity, you can assume $n$ to be a power of two.

Algorithm Loopy ($n : \text{integer}$)
\begin{itemize}
    \item $i \leftarrow 1$
    \item \textbf{while} $i \leq n$ \textbf{do}
        \begin{itemize}
            \item \textbf{for} $j \leftarrow i \text{ to } 2i - 1$ \textbf{do}
                \begin{itemize}
                    \item print “Hi”
                \end{itemize}
            \end{itemize}
        \end{itemize}
    \end{itemize}
$i \leftarrow 2i$

Answer:
Problem 3. (10 points)

Imagine that you are facing a high wall that stretches infinitely in both directions. There is a door in the wall, but you don’t know how far away is the door or in which direction. It is pitch dark, but you have a very dim lighted candle that will enable you to see the door when you are right next to it.

1. Show an algorithm that enables you to find the door by walking at most $O(n)$ steps in the worst case, where $n$ is the number of steps that you would have taken if you knew where the door is and walked directly to it (note that your algorithm does not know the value of $n$ in advance)

2. What is the constant multiple\(^1\) in the worst-case analysis for your algorithm?

Answer:

\(^1\)Observe that if you knew $n$, you could trivially solve the problem in $3n$ steps in the worst case (since you do not know the direction of the door); an interesting question (beyond the scope of this problem) is: what the minimum constant $c > 3$ that can be achieved by a linear-time algorithm (in the worst case)?
Problem 4. (10 points)

Given the following recurrence relation

\[ T(n) = \begin{cases} 1 & n = 1 \\ 4T\left(\frac{n}{2}\right) + 3 & n > 1 \end{cases} \]

1. Solve it exactly (i.e., without using any asymptotic notation) by iterative substitutions

2. Prove by induction that your solution is correct

Answer: