HW2: Algorithm Analysis – 20 pts
This week's lab will cover the concepts of algorithm analysis discussed in lecture.

Due Date
Friday 2020/2/10 at 4:00pm

Instructions
Questions below marked with an asterisk (*) are group problems. You may speak to anyone in class about the problem, ask questions, or help your classmates. However, you must write up and submit your own solution to the problem.

All other questions are intended as individual work. You may not discuss them with anyone other than the Instructor or the TAs.

Problems

1. (4 pts.) Use a step-counting approach to find an expression for the amount of "work" the following code performs. Use the rules of simplification to find a tight upper bound in terms of big-oh. Show that your tight upper bound meets the definition of big-oh. Assume n is the size of the input.

```
int x = 0;
for (int i = 1; i <= n; i++) {
     for(int j = 1; j <= i; j++) {
          x++;
     }
}
```

2. (4 pts.) Suppose a particular algorithm has time complexity $T(n)=n^2$, and that executing an implementation of it on a particular machine takes t seconds for n inputs. Now suppose we are presented with a machine that is 64 times as fast. How many inputs could we process on the new machine in t seconds? Be sure to show your work to justify your answer.

3. *For the following step-wise definition of *T(n)*:

$$T(n)=\begin{cases} T(n-1)+6 & \text{for } n>1 \\ 5 & \text{for } n=1 \end{cases}$$

   a. (4 pts.) Use the "find the pattern" approach to find the closed form of *T(n)*.
   b. (4 pts.) Use induction to prove that your closed form is equivalent to the recurrence relation.

4. (4 pts.) The power set of a set *S*, denoted $\wp(S)$, is the set of all subsets of *S*. Use induction to prove that the size of the power set of S with n elements (that is, for $|S|=n$) is given by $|\wp(S)|=2^n$.