CS 325
Due: Friday 18 Jan

Homework #1
Running Times

1. Show that $\Theta(\log n)$ is a reasonable notation by showing that for every base $b > 1$,

$$\log_b n \; = \; \Theta(\log_2 n).$$

2. Show that $\sum_{i=1}^{n} i^{17} \; = \; \Theta(n^{18})$.

3. Show that $n! \; = \; \mathbb{O}(n^n)$ but that $n^n \; \neq \; \Theta(n!)$.

4. Devise both a *recursive* algorithm and an *iterative* algorithm to compute Fibonacci numbers, and predict the run time of each of these two algotithms to big Θ order.

5. Plot the following 3 types of functions on the following 2 types of plots:

   (a) An exponential function (like 2^n)
   (b) A polynomial function (like $10n^2$)
   (c) A bounded function (like $10 + \sin n$)

   (a) a semi-log plot (X=linear scale, Y=logarithmic scale)
   (b) a log–log plot (X=logarithmic scale, Y=logarithmic scale)

   What **regularities** do you find in these plots?

6. Plot the following empirical data values:

   Input Size (first row) Execution Time (second row)

12	13	14	15	16	17	18	19	20	21	22	23	24
0.028	0.055	0.109	0.165	0.330	0.660	1.32	2.58	5.05	10.2	20.3	40.7	81.4

   What is the best kind of graph to use for this data set and why?
   What would you guess the complexity of this function is and why? (for instance: linear, polynomial, exponential, etc.)

7. What is the following algorithm computing?
   Choose an inductive variable and state an inductive hypothesis you would use to prove that this algorithm does the job you claim it does.

```
WHILE  B  >  0
   Subtract 1 from  B
   Add      1 to    A
ENDWHILE
```

8. Use the data in the following table to decide if the run time has the form $\Theta(2^n)$. Use this data to predict the run time for $n = 30$.

n	Time (ms)	$T(n)/T(n-1)$
5	10	
6	30	3
7	230	7.7
8	250	1.1
9	1287	5.1
10	1810	1.4
11	4270	2.4
12	10471	2.5
13	19398	1.9
14	39447	2.0
15	77669	2.0
16	147832	1.9
17	301652	2.0

Run Times for Prog-A and Prog-B as functions of n				
n	Prog-A Time (ms)		n	Prog-B Time (ms)
2	0		2	0
4	0		4	0
8	0		8	0
16	0		16	0
32	0		32	15
64	0		64	31
128	15		128	78
256	31		256	218
512	140		512	625
1024	500		1024	1875
2048	2000		2048	5578
4096	8000		4096	16812
8192	32000		8192	50438
16384	128094		16384	151578
32768	512376		32768	454734

9. Use the data in the above table for Prog-A and Prog-B. Plot this data and decide if the run times can be reasonably be written in the form $\Theta(n^k)$. Use you plot to estimate a k_A for Prog-A and a k_B for Prog-B. Which program do you expect to be asymptotically faster?

Important note from the grader: Besides getting the "right answers", you will want to make sure your work is *neat* and *clear*. When doing graphs, for example, this means:

- labeling your graph with a title or description,
- labeling and numbering your axes, and
- choosing appropriate aspect ratios.

The graph should be computer generated. The labeling may be done by computer or by hand.