. **Exercises on Chapter 1**

**Standard instructions for exercises**: try as many as you have time after reading the chapter once. Note: the more you do, the better you will do in exams. You may work as a team. Bring a **readable solution to one(1)** exercise to class with you on a piece of paper. For example: 3g is enough for one class. **Include the names** of the people who worked on it and a **copy of the problem**. Put it on the teacher's desk before the class starts. You may be asked to present it. I will grade it, comment on it, and return it by the following class.

**Note**: We will use the `big-O` notation and theory of graphs in last part of this class. These exercises will review these topics for you. Exercise 3 for example covers the kind of facts that we will use extensively in the last 6 sessions of this class.

1. Prove that if $x \le 8$ then $2^x \ge 4x^2$. (hint: page 21 Example 1.17)

! 2. Prove for all $c \ge 1$, there is an $x_0$ such that if $x \ge x_0$ then $2^x \ge cx^2$.

3. Find a definition of the big-O notation and use it to prove these O(.) facts -- assume f, g, h ... are non-negative functions mapping integers into real numbers:

(a): f(n) is O(f(n)).

(b): If f(n) is O(g(n)) and g(n) is O(h(n)) then f(n) is O(h(n)).

(c): For constant a, a*f(n) is O(f(n)).

(d): If for all n( f(n) $\le$ g(n) ) then f(n) is O(g(n)).

(e): If f(n) is O(g(n)) then f(n)+g(n) is O(g(n)).

(f): For all natural numbers p and q, if $p \le q$ then $n^p$ in $O(n^q)$ . Hint. Use d above.

! (g): Use a..f above and induction to prove that if f(n) is a polynomial with highest powered term of form $an^p$ then f(n) is $O(n^p)$.

4. Go to the course web site and find a C++ program called `time1.cpp`. Read it and think about what it does. Download or save a copy and compile and run it. Test it with several input values. Write up your results and

feelings on this experiment.

5. A graph has a set of nodes N and a set of edges E. Each edge is a pair $\{n_1, n_2\}$ of nodes. You can check out the various definitions of graph on the web.

a. How many edges can there be if there are n nodes? Prove your formula by induction.

b. If $\{n_1, n_2\}$ is in E then $n_1$ is connected to $n_2$. We say that $(n_1, n_2)$ is a **path** of length 1 connecting $n_1$ to $n_2$. A path of length *p* is a list of *p*+1 nodes $(n_1, n_2, ..., n_{p+1})$ where each pair of nodes is connected: $\{n_i, n_{i+1}\}$ is in E for all i: 1..p. A **connected graph** has a path from every node to another node. A **simple path** has no repeated nodes. A **cycle** has equal first and last nodes. A **simple cycle** has no other repeated nodes except the first and last. A cycle is **Hamiltonian** if it is simple and has every node in it. Draw a graph with 8 nodes with a Hamiltonian cycle. First do it the easy way: start with the cycle and then add some extra edges. Then draw a connected graph with 8 nodes and at least 15 edges and try to find a Hamiltonian cycle in it.

c. The **degree of separation** of two nodes in a graph is the length of the shortest path starting at one and ending at another. Prove that this shortest path must be simple. Bonus: Research the topic of degrees of separation on the web. Prepare and present a 3 minute presentation of what you find.

! d. Draw a connected graph with 8 nodes and calculate for every pair of nodes their degree of separation. How long does it take to find out if a given node is with 6 degrees of separation from every other node? Suppose the graph had 20 nodes how long would it take? How would you find the **largest degree of separation** in a large graph? An example graph: Nodes = `all actors who have played a part in a movie`. Edges=`Two actors who have played parts in the same movie`. How many Nodes? Edges? How would you verify that **Kevin Bacon** has 6 degrees of separation from every other actor? How would you verify that no other actor is equally central to this graph?