Solutions for Homework 1

f(n)	$\log^k n$	n^k	$\sqrt{n}$	$\log(n!)$
g(n)	n^ϵ	c^n	$n^{\sin n}$	$\log(n^n)$
Is $f(n)$ $O(g(n))$?	Yes	Yes	No	Yes
Is $f(n)$ $\Omega(g(n))$?	No	No	No	Yes
Is $f(n)$ $\Theta(g(n))$?	No	No	No	Yes
Is $f(n)$ $o(g(n))$?	Yes	Yes	No	No

Table 1:

Problem 1: (Grade 16 pts): In Table 1, $k \geq 1, \epsilon > 0, c > 1$. Please answer yes or no, and also justify your answer in each case.
Column1:
We will prove that $\log^k n = \mathrm{o}(n^\epsilon)$.
We want to show that for any constant $c > 0$, there exists a constant $n_0 > 0$:

$$\log^k n < cn^\epsilon.$$

Take the log of both sides

$$\log \log^k n < \log cn^\epsilon$$

$$k \log \log n < \log c + \epsilon \log n$$

This is true for large enough n (it can be proven using L'Hopital's rule).
So, $\log^k n = \mathrm{o}(n^\epsilon)$. (1)
From (1), we conclude that:

$$\log^k n = O(n^\epsilon). \ (2)$$

$$\log^k n \neq \Theta(n^\epsilon). \ (3)$$

$$\log^k n \neq \Omega(n^\epsilon). \ (4)$$

Column2:
We will prove that $n^k = o(c^n)$.
We want to show that for any constant $a > 0$, there exists a constant $n_0 > 0$:

$$n^k < ac^n$$

$$\log n^k < \log ac^n$$

$$k \log n < \log a + \log cn$$

This is true for large enough n (it can be proven using L'Hopital's rule).
So, $n^k = o(c^n)$. (5)

From (5), we conclude that:

$$n^k = O(c^n). \ (6)$$
$$n^k \neq \Theta(c^n). \ (7)$$
$$n^k \neq \Omega(c^n). \ (8)$$

Column3:
Observe the function n^{sinn}:
The value of the exponent is oscillating between 1 and -1, taking all values in between. So, the function oscillates between $\frac{1}{n}$ and n. $\sqrt{n}$ and n^{sinn} cannot be compared using asymptotic notation. The limit $\frac{\sqrt{n}}{n^{\text{sinn}}}$ oscillates.

Column4:

$$n! \leq n^n$$
$$\Rightarrow \log n! \leq \log n^n$$

So, picking up $c = 1, n_0 = 1$ we prove that $\log n! = O(\log n^n)$ (9)

The following bound holds for all n:

$$n! \geq \sqrt{2\pi n}(\frac{n}{e})^n$$
$$\Rightarrow n! \geq (\frac{n}{e})^n$$
$$\Rightarrow \log n! \geq \log(\frac{n}{e})^n$$
$$\Rightarrow \log n! \geq n \log \frac{n}{e}$$
$$\Rightarrow \log n! \geq n(\log n - \log e)$$
$$\Rightarrow \log n! \geq \frac{nlogn}{2}$$
$$\Rightarrow \log n! = \Omega(\log n^n) \ (10)$$

From (9),(10) we conclude:

$$\log n! = \Theta(\log n^n)$$
$$\log n! \neq o(\log n^n)$$

Problem 2: (Grade 6 pts) State true or false. Justify your answer (Give reasons if your answer is true, give a counter-example if your answer is false). Assume throughout that $f(n) \geq 0$ and $g(n) \geq 0$ for all n.

1. If $f(n)$ is $O(g(n))$ then $\log f(n)$ is $O(\log g(n))$. Assume that $\log(g(n)) > 0$ and $f(n) \geq 1$ $\forall n$.
   It's TRUE.
   Proof:
   We have that $f(n)$ is $O(g(n))$:

$$f(n) \leq cg(n)$$

$$\Rightarrow \log f(n) \leq \log c + \log g(n)$$
$$\Rightarrow \log f(n) \leq c' \log g(n)$$
$$\Rightarrow \log f(n) = O(\log g(n))$$

Here, we made the assumption that $g(n) \to \infty$ as $n \to \infty$.
For the other cases, if you are interested contact the instructor or the TA.

2. $f(n) + o(f(n))$ is $\Theta(f(n))$.
   TRUE
   Proof:
   Let $g(n) = o(f(n))$. For every c, there exists n_0:

$$g(n) < cf(n) \ for\ every\ n \geq n_0$$

$$\Rightarrow g(n) + f(n) < (c+1)f(n)$$
$$\Rightarrow f(n) \leq g(n) + f(n) < (c+1)f(n)$$
$$\Rightarrow f(n) \leq o(f(n)) + f(n) \leq (c+1)f(n)$$
$$\Rightarrow f(n) + o(f(n)) = \Theta(f(n))$$

3. $f(n)$ is $O\left((f(n))^2\right)$.
   TRUE
   If $f(n) \geq 1$ the statement holds, since $f(n) \leq f(n)^2$ under that assumption. So, picking up $c = 1$ and $n_0 > 0$ we are done.
   For the other case, if you are interested contact the instructor or the TA.

Problem 3: (Grade 3) Prove that $F_N \geq 2^{N/2}$ for all $N \geq 2$. Here, $F_0, F_1, F_2, \ldots$ are the Fibonacci numbers.

(a) Base case: It holds for $N = 2 : F_2 = 2 \geq 2^{2/2} = 2$.
(b) Induction hypothesis: Suppose it holds for $N = 2, ..., k$.
(c) Inductive step: Prove that it holds for $n = k + 1$.

$$F_{k+1} = F_k + F_{k-1} \geq 2^{k/2} + 2^{(k-1)/2}$$
$$\Rightarrow F_{k+1} \geq 2^{(k-1)/2} + 2^{(k-1)/2}$$
$$\Rightarrow F_{k+1} \geq 2^{((k-1)/2)+1}$$
$$\Rightarrow F_{k+1} \geq 2^{(k+1)/2}$$

Problem 4: (Grade 8 pts) You have a list of n real numbers, and another number x. You need to find out whether the sum of any two consecutive numbers in the list equals x or not. Give an algorithm to solve the problem. Analyze its complexity. For full grade you need to give a $O(n \log n)$ algorithm.

The idea is the following:
Add element 1 and element 2. Compare the sum with number x.
If you found it STOP, else continue like this adding element i with element $i + 1$ till $i = n - 1$.
This requires $O(n)$ operations.

Now give an algorithm which finds out whether there are p consecutive elements whose sum equals x, where p is an input number. Analyze its complexity.

Follow similar logic as before:
Add elements 1 to p. If you found it STOP, else subtract element 1 and add element $p+1$ into the previous sum.
In general, subtract element i and add element $i+p$ to the previous sum. Again, this requires $O(n)$ operations because of the efficient method of keeping the intermediate result so that not to have redundancy.

Problem 5: (Grade 7) You have a real number x, and a sequence of real numbers $a_0, \ldots, a_{n-1}$. Give an algorithm to find out the value of the polynomial $\sum_{i=0}^{n-1} a_i x^i$. Analyze the complexity of your algorithm. For full grade you need to give a $O(n)$ algorithm. Please note that the basic operations are addition, multiplication, subtraction, division, memory access, read and write operations. In particular, i multiplications need i basic steps.
The logic is the following:
An efficient way can be constructed if you notice that x^i can be computed like this:

$$x^i = x^{i-1}x$$

So, store the result x^{i-1} so that in the next step you can compute x^i by multiplying the previous power by x.

The algorithm is like this:

$$poly = 0;$$

$$power = 1;$$

$$for(i = 0; i \leq n-1; i++)\{$$

$$poly = poly + a_i * power;$$

$$power = power * x;\}$$

This requires $O(n)$ operations.