Instructions: Please make sure that your solutions are well-thought-out, complete, concise, rigorous, and clearly written. Each solution sheet should be marked with your name, the course number, the homework number, the problem number and the date. You are allowed to discuss problems and exchange solution ideas with other students, but you must write up your own solutions. If you collaborated with other students on a problem, please acknowledge them in your write up. These are the fun problems to solve. Some of them require a bit of thinking. Please start early!

Problem 1: How many slices of pizza can a person obtain by making \( n \) straight vertical cuts with a pizza knife? Assume that the pizza is a convex two dimensional object (e.g., a circular pizza). Give a recurrence for the number of slices, and solve the recurrence to give a closed form solution. (15 Points)

Problem 2:
(a) Use mathematical induction to prove the Cassini’s identity:

\[ F_{n+1}F_{n-1} - F_n^2 = (-1)^n, \quad \text{for } n > 0, \]

where \( F_k \) is the \( k \)th Fibonacci number. (5 Points)

(b) Use (a) to show that \( F_{n+1}^2 - F_nF_{n+1} = (-1)^n. \) (5 Points)

(c) The Fibonacci number \( F_n \) can be defined for negative integer \( n \). Here is how they are defined.

\[
\begin{align*}
\text{n:} & \quad \ldots \quad -5 \quad -4 \quad -3 \quad -2 \quad -1 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad \ldots \\
\text{F_n:} & \quad \ldots \quad 5 \quad -3 \quad 2 \quad -1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 2 \quad 3 \quad 5 \quad \ldots \\
\text{Show by induction that } F_{-n} = (-1)^{n-1}F_n, \forall n \in \mathbb{Z}. \quad (5 \text{ Points})
\end{align*}
\]

(d) Does Cassini’s identity hold for all \( n \in \mathbb{Z} \)? (5 Points)

(e) Prove that \( \left( \begin{array}{cc}
1 & 1 \\
1 & 0
\end{array} \right)^n = \left( \begin{array}{cc}
F_{n+1} & F_n \\
F_n & F_{n-1}
\end{array} \right), \) where \( F_k \) is the \( k \)th Fibonacci number. (Hint: Use induction) (5 Points)

Problem 3: Rank the following functions by order of growth. Partition your list into equivalence classes such that \( f(n) \) and \( g(n) \) are in the same class if and only if \( f(n) = \Theta(g(n)) \). Give brief justifications. (21 Points)

\[
\begin{array}{ccccccc}
(\sqrt{2})^{lg n} & n! & 2^n & (lg n)^{lg n} & 4^{lg n} & 2^{2^n+1} \\
e^n & (n+1)! & lg(n!) & \frac{3^n}{2} & n^{1/lg n} & ln ln n \\
2^{1000} & n & n \cdot 2^n & ln n & 1 & n lg n
\end{array}
\]

Problem 4: Give asymptotic upper and lower bounds for \( T(n) \) in each of the following recurrences. Assume that \( T(n) \) is constant for sufficiently small \( n \). Make your bounds as tight as possible. Justify your answers. (24 Points)
(a) $T(n) = T(9n/10) + n$.
(b) $T(n) = 16T(n/4) + n^2$.
(c) $T(n) = T(\sqrt{n}) + 1$.
(d) $T(n) = T(\frac{n}{3} + \log n) + 1$.
(e) $T(n) = 2T(n - 1) + n^5$.
(f) $T(n) = 7T(n/2) + n^2$.
(g) $T(n) = 2T(n/2) + \log(n!)$.
(h) $T(n) = 2T(n - 1) + 1$.

**Problem 5:** One way to sort an array $A[p \cdots q]$ of $n$ numbers is to scan the array, and find the minimum ($\min$) and maximum ($\max$) elements. Then exchange the first element of the array $A[p]$ with $\min$ and the last element of the array $A[q]$ with $\max$. This procedure is then repeated for the array $A[p + 1 \cdots q - 1]$, and so on, until the entire array is sorted. Give a recursive algorithm to sort the array $A$ using the above idea. Write the recurrence for the running time of your algorithm, and solve the recurrence to give tight asymptotic bound on the running time. \hspace{1cm} (15 Points)