1. Prove or disprove:
   
   • (8 points) If $f(n)$ and $g(n)$ are non-negative integer functions of $n$ such that $f(n) = \Theta(g(n))$, then, $(f(n))! = \Theta((g(n))!)$. 
   
   • (8 points) $(\log n)^{\sqrt{n}} = O\left((\sqrt{n})^{\log n}\right)$. 

Note: You are supposed to give proofs to the time bounds of your algorithms. Read the questions carefully before attempting to solve them.
2. (17 points) Input is a sequence $X = k_1, k_2, \ldots, k_n$ of arbitrary real numbers. The problem is to identify an element of $X$ whose rank in $X$ is at most $\sqrt{n}$. Present a Monte Carlo algorithm for solving this problem. Your algorithm should run in $O(\sqrt{n} \log n)$ time. Show that the output of your algorithm will be correct with a high probability. (The rank of an element $x$ in a sequence $X$ is defined as $|\{q \in X : q < x\}| + 1$). You may use the following fact: $(1 - x)^{1/x} \leq 1/e$, for any $0 < x < 1$. 
3. (16 points) Recall that a heap on $n$ keys can be represented as an array $a[1 : n]$. A min-heap supports three operations: Find-min, Insert, and Delete-min. Show how we can perform the following operation also in a heap: $\text{Delete}(i)$: delete the element $a[i]$ from the heap. Your algorithm for this operation should run in time $O(\log n)$. 
4. (17 points) Present a data structure for real numbers that can perform the following operations:

   a) $\text{Insert}(x)$: Insert the element $x$ into the data structure;
   
   b) $\text{Search}(x)$: Check if $x$ is in the data structure;
   
   c) $\text{Delete}(x)$: Delete $x$ from the data structure; and
   
   d) $\text{Find}(i)$: Return the $i$th smallest key in the data structure.

Each operation should take $O(\log n)$ time. (*Hint:* Modify a 2-3 tree appropriately.)
5. Solve the following recurrence relations:

(a) (8 points)

\[ T(n) = \begin{cases} 
  1 & n \leq 2 \\
  256 \cdot T\left(\frac{n}{2}\right) + n^8 & n > 2 
\end{cases} \]

(b) (9 points)

\[ T(n) = \begin{cases} 
  1 & n \leq 2 \\
  \frac{1}{n} [T(n - 1) + T(n - 2) + \cdots + T(2) + T(1) + T(0)] & n > 2 
\end{cases} \]
6. (17 points) Input are \( m \) sorted sets \( S_1, S_2, \ldots, S_m \) with \( \sum_{i=1}^{m} |S_i| = n \). Present an \( O(n) \) time algorithm to compute the intersection of these \( m \) sets.