CSL 356: Problem Set 0

1. Sort the functions given below from asymptotically smallest to asymptotically largest, indicating ties if there are any.
$$\lg((\sqrt{n})!), \lg(\sqrt{(n!)}), \sqrt{\lg(n!)}, (\lg(\sqrt{n}))!, (\sqrt{(\lg n)})!, \sqrt{(\lg n)!}$$

2. Solve the following recurrences. State tight asymptotic bounds for each function in the form $\Theta(f(n))$ for some standard mathematical function $f(n)$.

   (a) $A(n) = 2A(n/4) + \sqrt{n}$

   (b) $B(n) = 3B(n/3) + n/\lg n$

   (c) $C(n) = \frac{2C(n-1)}{C(n-2)}$

   (d) $D(n) = D(n-1) + 1/n$

3. We are given 10 distinct numbers $a_1, \ldots, a_5$ and $b_1, \ldots, b_5$. We know that $a_1 < a_2 < a_3 < a_4 < a_5$ and $b_1 < b_2 < b_3 < b_4 < b_5$. Our aim is to find the median of these 10 numbers (i.e., the 5-th smallest), using as few key comparisons in the worst-case as possible.

   - Draw a decision tree for one such method.
   - What is the worst-case number of comparisons made by your decision tree?

4. The n^{th} Fibonacci binary tree $\mathcal{F}_n$ is defined recursively as follows:

   - $\mathcal{F}_1$ is a single root node with no children.
   - For all $n \geq 2$, $\mathcal{F}_n$ is obtained from $\mathcal{F}_{n-1}$ by adding a right child to every leaf and adding a left child to every node that has only one child.

   (a) Prove that the number of leaves in $\mathcal{F}_n$ is precisely the n^{th} Fibonacci number: $F_0 = 0, F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for all $n \geq 2$.

   (b) How many nodes does $\mathcal{F}_n$ have? For full credit, give an exact, closed-form answer in terms of Fibonacci numbers, and prove that your answer is correct.

   (c) Prove that the left subtree of $\mathcal{F}_n$ is a copy of $\mathcal{F}_{n-2}$.