CSE 332: Data Structures and Parallelism

Section 3: BSTs, Recurrences, and Amortized Analysis

0. Interview Question: Binary Search Trees

Write pseudo-code to perform an in-order traversal in a binary search tree without using recursion.

1. Big-Oh Bounds for Recurrences

For each of the following, find a Big-Oh bound for the provided recurrence.

(a) $T(n) = \begin{cases} 1 & \texttt{if } n = 1 \\ 8T(n/2) + 4n^2 & \texttt{otherwise} \end{cases}$

(b) $T(n) = \begin{cases} 1 & \texttt{if } n = 1 \\ 7T(n/2) + 18n^2 & \texttt{otherwise} \end{cases}$

(c) $T(n) = \begin{cases} 1 & \texttt{if } n = 1 \\ T(n/2) + 3 & \texttt{otherwise} \end{cases}$

2. Recurrences and Closed Forms

For the following code snippet, find a recurrence for the worst case runtime of the function, and then find a closed form for the recurrence.

Consider the function g:

```
g(n) {
   if (n <= 1) {
      return 1000;
   }
   if (g(n/3) > 5) {
      for (int i = 0; i < n; i++) {
         System.out.println("Yay!");
      }
      return 5 * g(n/3);
   }
   else {
      for (int i = 0; i < n * n; i++) {
         System.out.println("Yay!");
      }
      return 4 * g(n/3);
   }
}
```

- Find a recurrence for $g(n)$.

- Find a closed form for $g(n)$.

3. MULTI-pop

Consider augmenting the Stack ADT with an extra operation:

> `multipop(k)`: Pops up to k elements from the `Stack` and returns the number of elements it popped

What is the amortized cost of a series of `multipop`'s on a `Stack` assuming `push` and `pop` are both $\mathcal{O}(1)$?

4. MinVL Trees

Draw an AVL tree of height 4 that contains the minimum possible number of nodes.

5. AVL Trees

Insert 6, 5, 4, 3, 2, 1, 10, 9, 8, 7 into an initially empty AVL Tree.

6. AVL Trees

Given a binary search tree, describe how you could convert it into an AVL tree with worst-case time $\mathcal{O}(n\lg(n))$. What is the best case runtime of your algorithm?

7. HeapVL Trees

Is there an AVL Tree that isn't a heap? Is there a heap that isn't an AVL tree? Is there a binary search tree that is neither? Is there a binary search tree that is both?

8. B-Trees

(a) Insert the following into an empty B-Tree with $M = 3$ and $L = 3$: $12, 24, 36, 17, 18, 5, 22, 20$.

(b) Delete $17, 12, 22, 5, 36$

(c) Given the following parameters for a B-Tree with $M = 11$ and $L = 8$

- Key Size = 10 bytes
- Pointer Size = 2 bytes
- Data Size = 16 bytes per record (includes the key)

Assuming that M and L were chosen appropriately, what is the likely page size on the machine where this implementation will be deployed? Give a numeric answer and a short justification based on two equations using the parameter values above.