CS 240: Data Structures and Data Management **Spring 2024**

Tutorial 08: June 28

1. (Question 2c) Prove that $\sqrt{\log(\log(n))} \in o(n)$ from first principles. Here, you can assume that we limit the domain of our functions to $n \geq 2$, so that $\sqrt{\log(\log(n))}$ is well-defined.

2. (Question 3a) What is the average runtime, as a function of n, of the following pseudo-code segment? The average is taken over all choices of (A, i), where A is a size-n array storing the integers $\{0, 1, ..n-1\}$ and i is in $\{0, \ldots, n-1\}$. Give a $\Theta(\ )$ expression. Justify your answer.

```
1 Procedure algo(A, i)
    Input: A: an array of size n storing all integers in the set {0, 1, ..., n − 1}
    Input: i: integer in {0, ..., n − 1}
2   if A[0] = i then
3     for j ← 1 to i^2 do
4       print j
5     end
6   end
```

3. (Question 5a) Design a comparison-based algorithm that does the following. You are given a size n-array of distinct integers A and a binary tree T with n nodes. You have to assign each integer $A[i]$ $(i = 0, \ldots, n-1)$ to one of the nodes in T, so that the resulting tree is a binary search tree. You can assume that the nodes in T have attributes `key` (initially un-initialized), `left` and `right`, and you can only change the value of the `key` attribute (if z is a node, you can write z.`key`, etc). You can modify A if you want to. You can assume that you have a helper function `is-empty`(T) that tests if a binary tree T is the empty binary tree.

Briefly justify correctness of your algorithm, and analyze its worst case runtime in terms of n and the height h of T (we only ask for a big-O). For full credit, your algorithm should run in worst case time $O(nh)$.