CS120: Intro. to Algorithms and their Limitations	Prof. Salil Vadhan
Midterm	
Harvard SEAS - Fall 2021	*Date Thurs. Oct. 14, 2021*

0 Instructions

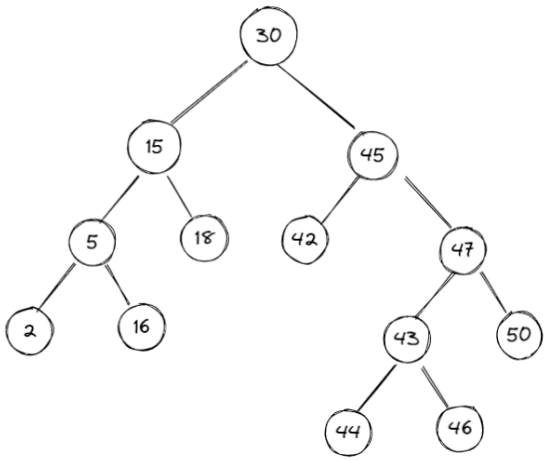
- Be sure to write your name and Harvard ID on every solution page.
- This is a closed-book exam.
- Write in pen.
- The purpose of the midterm is synthesize the material so far and help you identify where there are gaps to fill. Thus it will be graded on an N/L/R/R* scale, and revisions to convert an L to an R will be possible (using your allotted revisions). If you do not manage to entirely solve a problem, do describe your understanding of the concepts in the problem and how one might approach it. A partial answer that shows your understanding is better than a "complete" solution that has conceptual flaws.
- There are four problems:
  - Problem 1 is a short-answer problem about 3 concrete examples.
  - Problem 2 is a series of 3 true/false questions on a variety of different topics. For each, you should justify your answer in one or two sentences.
  - Problem 3 asks you to identify the best algorithms or data structures for 3 different application scenarios. For each, you should describe how you would use the algorithm or data structure and what the runtime would be as a function of the size of the population.
  - Problem 4 asks you to design an algorithm for a new problem by reduction to computational problems we have studied to class.

Name: _____ **HUID:** _____

1 Binary Search Trees

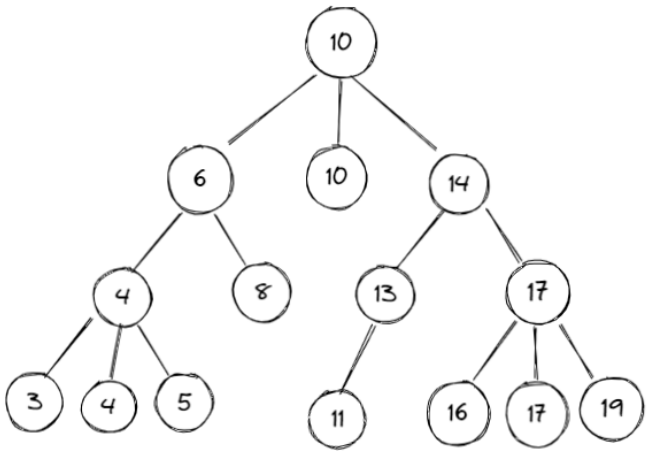
Consider the following three trees labelled with keys. One of the three is a valid Binary Search Tree. Identify which one is the the valid Binary Search Tree. On that one, show the tree after the operation Insert(16). (This is an ordinary BST insert, without worrying about balance or augmentation.) On the other two trees, point out which property of the BST property is violated.

1.1 Tree 1



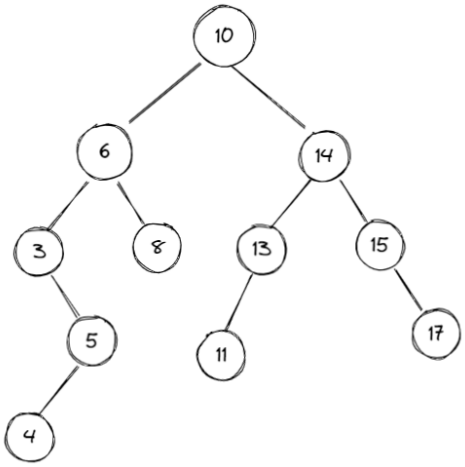
Name: _____ **HUID:** _____

1.2 Tree 2



Name: _____ **HUID:** _____

1.3 Tree 3



Name: _____ **HUID:** _____

2 True or False

State whether each of the following 4 statements are True or False, and justify your answer in one or two sentences.

2.1 Asymptotic Notation

There are functions $f, g : \mathbb{N} \to \mathbb{N}$ such that $f = \Theta(g)$ and $g = o(f)$.

2.2 Computational Problem

For every computational problem $\Pi = (\mathcal{I}, f)$, if A and B are algorithms that correctly solve Π, then for all inputs $x \in \mathcal{I}$, we have $A(x) = B(x)$.

Name: _____ **HUID:** _____

2.3 RAM Programs

If a computational problem can be solved in time $T(n)$ in the Word-RAM model, then it can be solved in time $O(T(n))$ in the RAM model. (For simplicity, you may restrict attention to Word-RAM programs that never overflow, i.e. all operations yield numbers smaller than 2^w where w is the current word size, and never try to access a memory location out of bounds. Dealing with those issues is an optional challenge if you have extra time!)

Name: _____ **HUID:** _____

3 Choosing Algorithms and Data Structures

Suppose the US Census Bureau was going to develop a new database to keep track of the exact ages of the entire US population, and publish statistics on it. The data it has on each person is an exact birthdate `bday` (year, month, and date) and a unique identifier `id` (e.g. social security number — pretend that these are assigned at birth).

For each of the three scenarios below,

1. Select the best algorithm or data structure for the Census Bureau to use from among the following:
   - sorting and storing the sorted dataset
   - storing in a binary search tree (balanced and possibly augmented)
   - storing in a hash table
   - running randomized QuickSelect.
2. Explain briefly how you would use the algorithm or data structure.
3. State what the runtime would be as a function of the size of the population or the number of updates.

In each scenario, you should assume (unrealistically) that the described queries or statistics are the *only* way in which the data is going to be used, so there is no need to support anything else.

3.1 Reporting Age Rankings

Every ten years as part of the Decennial Census, the Bureau collects a fresh list of (`id`, `bday`) pairs from the entire US population. (It does not reuse data from the previous Decennial Census, so everyone is re-surveyed.) In order to incentivize participation, the Bureau promises to tell every respondent their age-ranking in the population after the survey is done (e.g. "you are the 796,421'th oldest person among those who responded to the Census").

Name: _____ **HUID:** _____

3.2 Daily Quartiles

After each day, the Bureau obtains a list of (`id`, `bday`) pairs to add to or remove from its database due to births, deaths, and immigration, and publishes an updated 25th, 50th, and 75th percentile of the population ages.

3.3 Age Lookups

For privacy reasons, the Bureau decides to not publish any statistics on the population ages, but just wants to maintain a database where the age of any member of the population can be looked up quickly, and the database can be quickly updated daily according to births, deaths, and immigration.

Name: _____ **HUID:** _____

4 Reductions

For a dataset $x = (x_0 \ldots, x_{n-1})$ of real or rational numbers, the *mean* of x is defined to be $(\sum_{i=0}^{n-1} x_i)/n$. The mean can be too sensitive to outliers; a more robust alternative is the *trimmed mean*, where we drop the bottom 5% and top 5% of numbers from the dataset and take the mean of the resulting $.9n$ numbers.

By giving a reduction to a problem we have studied, explain how a trimmed mean of n rational numbers can be computed by a Las Vegas randomized algorithm in expected time $O(n)$. You should provide pseudocode for your algorithm/reduction and analyze its runtime. You may assume that n is divisible by 20 to avoid rounding issues in determining the bottom and top 5% of numbers. If you are unable to obtain an $O(n)$-time randomized algorithm, give the asymptotically fastest (randomized or deterministic) algorithm you can find.