CS240E W25	**Tutorial 4**	Jan. 31

Overview

- Amortized Analysis with Potential Functions
- Balanced Trees
- AVL Trees and Rotations
- Scapegoat Trees

Problems

Q1. Binary Counter.

An *n-bit binary counter* counts upward from zero and is stored as an array of n bits (the leftmost bit is least significant). It supports the operation *increment*, which adds 1 to the counter:

```
increment(A[0..n-1]):
    i = 0
    while(A[i] != 0):
        A[i] = 0
        ++i
    A[i] = 1
```

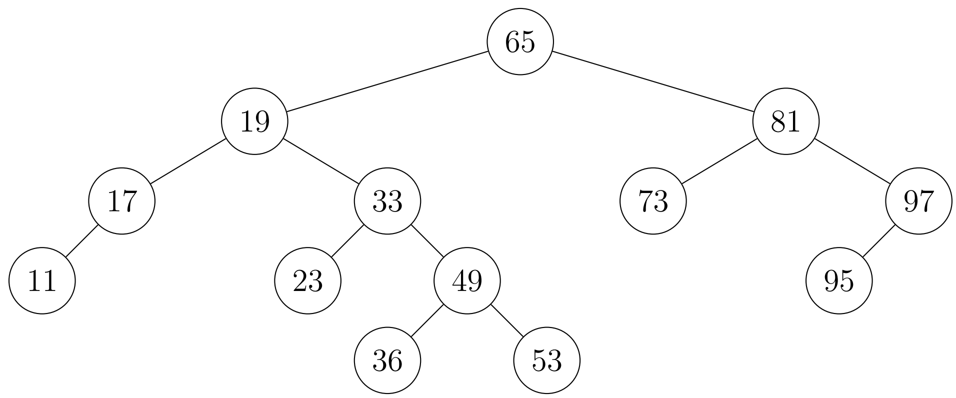
The running time for *increment* is $\Theta(k)$, where k is the final value of variable i, which is $\Theta(n)$ in the worst case. Show the amortized cost per *increment* of $\Theta(1)$.

Q2. 2-AVL Trees.

Define a 2-AVL tree to be a binary search tree where for every node, the difference of heights of its left and right subtree is at most 2. Prove that a 2-AVL tree has height at most $3 \log n$ where n is the number of nodes in the tree.

Q3. AVL Rotations.

Consider the AVL Tree shown below and perform `insert(61)`, then `delete(73)`.



Q4. Scapegoat Tree Reconstruction.

Insert the key 13 in the following scapegoat (2/3)-tree. The numbers in the brackets are the number of nodes in that subtree.