Assignment 2: Restaurant Ruckus

Due (on Ed) on Thu 22 Sept, 23:59

There are 4 types of creatures in Middle Earth: Dwarves, Hobbits, Elves, and Humans. You want to open a restaurant in Hobbiton, and soon realise the 4 types have very specific rules about who they accept to dine with, leading to... interesting occupancies in the restaurant area of your inn. Namely, you have tables which can accommodate up to 7 people: when a new customer arrives, they choose their table according to the following rules.

- Hobbits will always go to the **least crowded table**, and if they are all full will start a new table.
- Elves will choose to go to the closest table [to the door] **with only elves;** if there is none, they will start a new table.
- Dwarves will go to the closest table **with the fewest elves**, and if they are all full will start a new table.
- Humans will go to the closest table **as long as it is not dwarf-only, elf-only, or hobbit-only,** and if they are all full will start a new table.

This is a bit of a nightmare to handle for you, so to keep track you want to design a data structure (a type of tree! Elves love trees. 🧝 🌳) with the following properties:

- Each node corresponds to a table
- Given a node (table), you can:
    - Check if the table is full: is_table_full()
    - Check if the table is currently elves-only: is_elves_only()
    - Check if the table is currently dwarves-, hobbit-, or elves-only: is_dhe_only()
    - Check the number of people currently at the table: get_total_diners()
    - Check the number of elves currently at the table: get_elves()
    - Add an Elf, a Human, a Dwarf, or a Hobbit to a table (if the table is full, should return null): add_elf(), add_human(u), add_dwarf(), add_hobbit() *according to the above rules*.
    - Check the distance to the entrance of the inn (if there currently are n tables, returns a number between 1 (closest table) and n (farthest): get_distance()
- You can start a new table (originally empty) at the farthest position from the door: start_new_table()
- You can retrieve the least crowded table: get_least_crowded_table() [in case of ties, return the closest from the door] [should run in time O(1)]
- You can retrieve the current number of tables: get_number_tables() [should run in time O(1)]
- You can retrieve the number of customers: get_number_diners() [should run in time O(1)]
- You can retrieve the number of Elves, Dwarves, Humans, and Hobbits: get_number_elves(), get_number_humans(), get_number_dwarves(), get_number_hobbits() [should run in time O(1)]

Try to make all operations as efficient as possible!