Homework 4

1. You are writing a scheduler where events are marked with timestamps for when they will occur. You have two primary operations: schedule a future event and find the next occuring event. What data structure would you use?
2. Draw the tree for inserting 2, 4, 6, 8, 1, 3, 5, 7, 9 into an initially empty splay tree. Then delete 2 and redraw the tree.
3. Write a function to sort a list using a heap. Which of the sorting methods covered in class is it a variation on?
4. How many positions of a min-heap with n items do you need to examine in order to identify the maximum item?
5. For a project, you need a max-heap, but your library only contains a min-heap. How can you use this structure to fullfill your needs?
6. Your friend thinks that they can implement a FIFO queue using a priority queue by assigning each item a key equal to the size of the queue when it is enqueued. Does their strategy work? Prove that it does or provide a counterexample.
7. Show how to implement a LIFO stack using a priority queue and a single integer variable.
8. Which hash-table collision schemes can tolerate a load factor above 1 and which cannot?
9. Draw an 11-entry hash table using the hash function $h(x) = (3x + 5) \bmod 11$, to hash the keys 42, 26, 19, 35, 88, 62, 58, 7, 19, 53, and 12, resolving collisions with chaining.
10. Draw the same hash table, except resolve collisions with linear probing.
11. How could you implement a `Set` class using a Python dictionary?
12. Python has an `OrderedDict` class, which has $O(1)$ expected search times, like a standard `dict`, but it guarantees its iterator reports items in FIFO order, like a queue. Describe how to implement such a structure.