DA5300: Data Structures for Data Science(Worksheet-1)

Questions are from the reference textbook "Data Structures and Algorithms in Python", GoodRich, Tamassia, GoldWasser

March 2024

1 Stacks and Queues

1. R-6.1 What values are returned during the following series of stack operations,if executed upon an initially empty stack? push(5), push(3), pop(), push(2), push(8), pop(), pop(), push(9), push(1), pop(), push(7), push(6), pop(), pop(), push(4), pop(), pop().

2. R-6.2 Suppose an initially empty stack S has executed a total of 25 push opera- tions, 12 top operations, and 10 pop operations, 3 of which raised Empty errors that were caught and ignored. What is the current size of S?

3. R-6.3 Implement a function with signature transfer(S, T) that transfers all elements from stack S onto stack T, so that the element that starts at the top of S is the first to be inserted onto T, and the element at the bottom of S ends up at the top of T.

4. R-6.4 Give a recursive method for removing all the elements from a stack.

5. R-6.5 Implement a function that reverses a list of elements by pushing them onto a stack in one order, and writing them back to the list in reversed order.

6. R-6.7 What values are returned during the following sequence of queue operations, if executed on an initially empty queue? enqueue(5), enqueue(3), dequeue(), enqueue(2), enqueue(8), dequeue(), dequeue(), enqueue(9), enqueue(1), dequeue(), enqueue(7), enqueue(6), dequeue(), dequeue(), enqueue(4), dequeue(), dequeue().

7. R-6.8 Suppose an initially empty queue Q has executed a total of 32 enqueue operations, 10 first operations, and 15 dequeue operations, 5 of which raised Empty errors that were caught and ignored. What is the current size of Q?

8. R-6.9 Had the queue of the previous problem been an instance of ArrayQueue that used an initial array of capacity 30, and had its size never been greater than 30, what would be the final value of the front instance variable?

9. C-6.18 Show how to use the transfer function, described in Exercise R-6.3, and two temporary stacks, to replace the contents of a given stack S with those same elements, but in reversed order.

10. C-6.20 Describe a non-recursive algorithm for enumerating all permutations of the numbers $\{1, 2, ..., n\}$ using an explicit stack.

11. C-6.23 Suppose you have three nonempty stacks R,S,and T. Describe a sequence of operations that results in S storing all elements originally in T below all of S's original elements, with both sets of those elements in their original order. The final configuration for R should be the same as its original configuration. For example, if R = [1, 2, 3], S = [4, 5], and T = [6, 7, 8, 9], the final configuration should have R = [1, 2, 3] and S = [6, 7, 8, 9, 4, 5].

12. Describe how to implement the stack ADT using a single queue as an instance variable, and only constant additional local memory within the method bodies. What is the running time of the push(), pop(), and top() methods for your design? Describe how to implement the queue ADT using two stacks as instance variables, such that all queue operations execute in amortized O(1) time. Give a formal proof of the amortized bound.

13. C-6.27 Suppose you have a stack S containing n elements and a queue Q that is initially empty. Describe how you can use Q to scan S to see if it contains a certain element x, with the additional constraint that your algorithm must return the elements back to S in their original order. You may only use S, Q, and a constant number of other variables.

14. C-6.28 In certain applications of the queue ADT, it is common to repeatedly dequeue an element, process it in some way, and then immediately en- queue the same element. Modify the ArrayQueue implementation to include a rotate() method that has semantics identical to the combination, Q.enqueue(Q.dequeue( )). However, your implementation should be more efficient than making two separate calls (for example, because there is no need to modify _size).

15. C-6.31 Suppose Bob has four cows that he wants to take across a bridge, but only one yoke, which can hold up to two cows, side by side, tied to the yoke. The yoke is too heavy for him to carry across the bridge, but he can tie (and untie) cows to it in no time at all. Of his four cows, Mazie can cross the bridge in 2 minutes, Daisy can cross it in 4 minutes, Crazy can cross it in 10 minutes, and Lazy can cross it in 20 minutes. Of course, when two cows are tied to the yoke, they must go at the speed of the

slower cow. Describe how Bob can get all his cows across the bridge in 34 minutes.

2 Lists

1. R-7.1 Give an algorithm for finding the second-to-last node in a singly linked list in which the last node is indicated by a next reference of **None**.

2. R-7.2 Describe a good algorithm for concatenating two singly linked lists L and

3. R-7.3 Describe a recursive algorithm that counts the number of nodes in a singly linked list.

4. R-7.4 Describe in detail how to swap two nodes x and y (and not just their con- tents) in a singly linked list L given references only to x and y. Repeat this exercise for the case when L is a doubly linked list. Which algorithm takes more time?