

Algorithms and Data Structures

Summer Semester 2022

For discussion on Tuesday, May 14, 2022

1. (GTG 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()`.
2. (GTG R-6.2) Suppose an initially empty stack `S` has executed a total of 25 push operations, 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. (GTG 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. (GTG 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.
5. Look at line 33 of the delimiters matching code at

https://github.com/mjwestcott/Goodrich/blob/master/ch06/match_delimiters.py

Explain why this code works, and why the seemingly simpler line

```
if c != S.pop():
```

would not work.

6. (GTG 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. (GTG 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. Modify the given implementation of the `ArrayQueue` class, see

https://github.com/mjwestcott/Goodrich/blob/master/ch06/array_queue.py

to obtain an implementation of the `Deque` abstract data type, cf. GTG Section 6.3.1.