GS Interview Study Group

Friday, November 10th, 2017

Engineering Library 605, 4:00pm - 6:00pm

1 CtCI 2.4, p.94, LinkedLists

$Partition$: Write code to partition a linked list around a value x, such that all nodes less than x come before all nodes greater than or equal to x. If x is contained within the list, the values of x only need to be after the elements less than x (see below). The partition element x can appear anywhere in the "right partition" it does not need to appear between the left and right partitions.

EXAMPLE:

Input: $3 \rightarrow 5 \rightarrow 8 \rightarrow 5 \rightarrow 10 \rightarrow 2 \rightarrow 1$ [partition=5]
Output: $3 \rightarrow 1 \rightarrow 2 \rightarrow 10 \rightarrow 5 \rightarrow 5 \rightarrow 8$

2 CtCI 3.3, p.99, Stacks and Queues

$StackofPlates$: Imagine a (literal) stack of plates. If the stack gets too high, it might topple. Therefore, in real life, we would likely start a new stack when the previous stack exceeds some threshold. Implement a data structure of $SetOfStacks$ that mimics this. $SetOfStacks$ should be composed of several stacks and should create a new stack once the previous one exceeds capacity. $SetOfStacks.push()$ and $SetOfStacks.pop()$ should behave identically to a single stack (that is, pop() should return the same values as it would if there were just a single stack).

FOLLOW UP (Bonus Question)
Implement a function $popAt(intindex)$ which performs a pop operation on a specific sub-stack.