README

NAME: TaylorVu

NETID: tpv170000

CLASS SECTION: CS 3345.006

ArrayStack

My solution approach for this particular stack was to make a simple array list. The key aspect was setting capacity because all arrays come with fixed capacity (3.2.1 from Data Structures and Algorithm Analysis in Java). So, in the program I fixed an initial capacity that can be enlarged when is necessary which allows for array implementation for the Backmasking interface. The upper-bound O(N) complexity would be dependent of the bulk of the implementation and would be O(1) .

ListStack

My solution approach for the ListStack is to make a list that has a series a node. I put ListStackNode as inner class. I put for each node the for the ListStack each element to be linked to a successor such as in a linear fashion. The links would traverse through the list with the idea for the node to remain linked to the previous node by making the last node being null. Another way to option that I would like to try was to make doubly linked list to try optimizing the stack. The upper-bound O(N) complexity would be always remaining at O(1).