Data Structures and Algorithms.
Problem solutions. Week 2.

Ulyana Chaikouskaya
u.chaikouskaya@innopolis.university
Innopolis University, B23-DSAI-03

January 31, 2024

1 Task 1

1.1 Statement

In [Cormen, Section 16.1], a stack with an extra operation Multipop is discussed. What is the total cost of executing n of the stack operations PUSH, POP, and MUltiPOP, assuming that the stack begins with k_0 objects and finishes with k_n objects? Provide brief justification (1-2 sentences).

1.2 Solution

Let's break down the total cost of executing n stack operations, including Push, Pop, and Multipop, starting with k_o objects and ending with objects, the difference $(k_n - k_0)$ represents the net change in the size of the stack. Cost of PUSH - $O(1)$, POP - $O(1)$, and MULTIPOP - $O(n)$, cause its cost is proportional to the number of elements.

$T(n) = n - (k_n - k_0) = n - k_n + k_0$

2 Task 2

2.1 Statement

A sequence of Push, Pop, and SAVE operations is performed on a stack. SAVE operation copies all the elements of the stack that have not been backed up before. To keep track of which elements have a backup, the stack is equipped with a pointer to the most recently pushed element with a backup. PUSH does not affect the pointer, POP only affects the pointer if it pointed to the top element (in this case the pointer will be updated to point to the new top element after POP), and SAVE copies all elements from the pointer to the top of the stack and updates the pointer to point to the top.

Perform amortised time complexity analysis using the accounting method for a sequence of PUSH, POP, and SAVE operations performed on an initially empty stack:

2.2 Solution

(a) Specify actual cost, amortized cost, and accumulated credit for each operation. Assume that n_i is the size of stack before operation and k_i is number of backed up elements in the stack.

operation	actual cost	amortized cost	credit
PUSH	1	3	2
POP	1	0	-1
SAVE	$n_i - k_i$	0	$k_i - n_i$

(b) Show that the total amortized cost of a sequence of n operations provides an upper bound on the total actual cost of the sequence.

credit = amortized cost – actual cost $\Rightarrow \sum_{i=1}^{n} \hat{c_i} \geq \sum_{i=1}^{n} c_i$

$\sum_{i=1}^{n}(credit + \hat{c_i}) \geq \sum_{i=1}^{n} c_i$ It is clear that $\sum_{i=1}^{n}(credit) \geq 0$.
(c) Write down the asymptotic complexity for a sequence of n operations.
Operations' costs $O(1)$ at the same time n complexity is $\Theta(n)$ therefore:
$T(n) = \Theta(n)$