

Aufgabe 4

(4 a)

Tabelle 1: Vorbedingungen und Nachbedingungen für Deque Funktionen

Funktion	Vorbedingungen	Nachbedingungen
Konstruktor <code>q.Deque(N)</code>	<code>type(N) == int</code> <code>N ≥ 1</code>	<code>q ∈ Deque</code> Kapazität von <code>q == N</code>
<code>q.size()</code>	<code>q ∈ Deque</code>	<code>q</code> wird durch die Funktion nicht verändert Rückgabewert entspricht der tatsächlichen Größe
<code>q.capacity()</code>	<code>q ∈ Deque</code>	<code>q</code> wird durch die Funktion nicht verändert Rückgabewert entspricht der tatsächlichen Kapazität
<code>q.push(x)</code>	<code>q ∈ Deque</code> <code>x</code> ist ein gültiger Typ und Wert (für die Implementation in Python spielt das keine Rolle, da eine Liste Variablen unterschiedlicher beliebiger Typen halten kann)	Das letzte Arrayelement hat den Wert <code>x</code> falls die Größe nicht der Kapazität entspricht, wird sie um eins erhöht
<code>x = q.popLast()</code>	<code>q ∈ Deque</code> <code>q</code> ist nicht leer	Länge von <code>q</code> wird um eins vermindert <code>x</code> hat den Wert des entfernten Elements
<code>x = q.popFirst()</code>	<code>q ∈ Deque</code> <code>q</code> ist nicht leer	Länge von <code>q</code> wird um eins vermindert <code>x</code> hat den Wert des entfernten Elements