UNIVERSITY COMPUTER SCIENCE • EXERCISE SHEET

5.2 Lists, stacks and queues

Name: _____ Class: _____ Date: _____ **Total: 24 marks**

KEY WORDS stack • queue

Objective

Price the ends of the linear structures and trace them.

1 Key knowledge and worked examples

■ The layouts

An array indexes in one computed step but slides on front inserts; a linked list splices cheaply but walks to index ✓. A stack serves one end; a queue has two ends ✓.

■ Worked example: append amortises, front never does

Doubling regrows make `append` amortized $O(1)$: each element is copied at most once overall ✓. But 100 front inserts move $100 + 99 + \ldots + 1 = 5\,050$ elements – $\Theta(n^2)$ ✓.

2 Practice

2.1 `push 4, push 9, push 2, pop, push 7` – give the two pops that a further `pop, pop` returns. [3]

2.2 State the number of element moves for n front-inserts into an initially empty array, as a formula and at $n = 200$. [3]

2.3 Enqueue `x, y, z`, dequeue twice – name the front element. [3]

3 Exam-style questions

3.1 A queue is built on a Python list using `append` for enqueue and `pop(0)` for dequeue.

State the cost class of each operation and of n paired enqueue-dequeue rounds. Rebuild it on `collections.deque` and state both costs again. Give the total class ratio between the two designs at $n = 10^4$, with the reason the list version loses. [8]

3.2 The bracket validator pushes opening brackets and matches each closer against the top of the stack.

Trace the scan of `{ [ ] ( ) }` listing the stack after each character. Trace `( [ ) ]` and name the character and the mismatch that fails. State the empty-stack case and why it also fails. [7]