UNIVERSITY COMPUTER SCIENCE • EXERCISE SHEET

5.7 Searching

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

KEY WORDS linear search • binary search

Objective

Run both searches by hand and respect the sorted precondition.

1 Key knowledge and worked examples

■ The two searches

Linear walks, $O(n)$, no preconditions; binary halves a sorted range – 16 slots resolve in at most $\lceil \log_2 17 \rceil = 5$ probes ✓. On the course array, finding 43 probes $7, 11, 9, 10$ ✓.

■ Worked example: the miss

Searching 64 probes $7(31), 11(47), 13(55), 14(59), 15(63)$ – five probes, then the empty range reports absent; linear walks all 16 ✓.

2 Practice

2.1 A sorted array of 32 slots: give the worst-case binary probe count. [3]

2.2 State the update that makes binary search loop forever. [3]

2.3 Give the safe midpoint formula and the failure it prevents in fixed-width languages. [3]

3 Exam-style questions

3.1 The sorted array is $[2, 6, 10, 14, 18, 22, 26, 30]$ (indices 0–7).

Search 22: give every probe index, the value read, and the range kept after each probe. Search 5 (absent): give the probe trail and the final empty range. Give both worst-case bounds (8 slots) and the exact linear cost of each query. [8]

3.2 An unsorted array of 10^6 values must answer 10^4 queries.

Price all-linear: total comparisons. Price sort-then-binary: the $\Theta(n \log_2 n)$ sort cost (value at $n = 10^6$), plus the 10^4 queries at at most $\lceil \log_2(10^6 + 1) \rceil$ probes each (value of the ceiling). State the crossover condition on the number of queries that justifies sorting first. [7]