10.1 Sorting and packing algorithms

Name: _____ Class: _____ Date: _____ **Total: 27 marks**

KEY WORDS quick sort • bin packing

Sorting and packing algorithms

Objective

Execute bubble sort, quick sort and bin-packing algorithms exactly, showing every pass, pivot and placement, and compare bin counts across algorithms (objectives 1.1–1.2).

1 Key knowledge and worked examples

Rules of execution. Bubble sort: list after EVERY pass; stop after a pass with no swaps. Quick sort: pivot = the middle item (glossary method); list after each partition with the pivot in place. Bin packing: first-fit, first-fit decreasing (sort first), full-bin – show each item's bin and give the final bin count. More bins than the total÷capacity lower bound signals checking, not error.

Worked example – quick sort (original). Sort 9, 2, 7, 5, 3, 4, 6. Pivot (position 4 of 7) = 5: less = 2, 3, 4; greater = 9, 7, 6 → 2, 3, 4, **5**, 9, 7, 6. Sublist 2, 3, 4 (pivot 3): 2, **3**, 4. Sublist 9, 7, 6 (pivot 7): 6, **7**, 9. Final: 2, 3, 4, 5, 6, 7, 9.

Worked example – bin packing (original). Capacity 10; items 7, 5, 4, 3, 6, 2 (total 27 so at least 3 bins). First-fit: 7→b1, 5→b2, 4→b2 (9), 3→b1 (10), 6→b3, 2→b3 (8): bins (7,3), (5,4), (6,2). First-fit decreasing: 7→b1, 6→b2, 5→b3, 4→b2 (10), 3→b1 (10), 2→b3 (7): bins (7,3), (6,4), (5,2). Both 3 bins here.

2 Practice

2.1 Bubble sort 5, 1, 4, 2, showing the list after each pass. [3]

2.2 Quick sort 1, 6, 3, 5, 7, 2, 9, stating each pivot and the list after each partition. [4]

2.3 Bin capacity 8; items 5, 3, 4, 2, 6, 1. Apply first-fit, showing each placement, and state the number of bins used. [3]

3 Exam-style questions

3.1 *(Original.)* A courier has parcels of weights 9, 4, 6, 2, 8, 5, 3, 7 (kg) and boxes of capacity 15 kg. (a) Use the first-fit algorithm to pack the parcels, listing the contents of each box in order. (b) Use the first-fit decreasing algorithm to pack them. (c) Show that the total weight implies at least three boxes, and comment on the two algorithms' results. [9]

3.2 *(Original.)* The list 1, 6, 3, 4, 8, 2, 9 is to be sorted into ascending order. (a) Use the quick sort algorithm, stating the pivot at each stage and the list after each partition. (b) State one advantage quick sort has over bubble sort on large lists. [8]

Solutions

2.1 Pass 1: 5>1 swap, 5>4 swap, 5>2 swap → 1, 4, 2, 5. Pass 2: 4>2 swap → 1, 2, 4, 5. Pass 3: no swaps → sorted, stop. Final: 1, 2, 4, 5.

2.2 Pivot (position 4 of 7) = 5: less = 1, 3, 2; greater = 6, 7, 9, giving 1, 3, 2, **5**, 6, 7, 9. Sublist 1, 3, 2 (pivot 3): items less than 3, in order, are 1, 2 and none are greater, so the sublist becomes 1, 2, **3**; the pair 1, 2 (pivot 1) gives **1**, 2. Sublist 6, 7, 9 (pivot 7): less = 6, greater = 9 – already sorted. Final list: 1, 2, 3, 5, 6, 7, 9.

2.3 5→b1, 3→b1 (8), 4→b2, 2→b2 (6), 6→b3, 1→b2 (7): bins (5,3), (4,2,1), (6) – 3 bins.

3.1 (a) First-fit (capacity 15): 9→b1, 4→b1 (13), 6→b2, 2→b2 (8), 8→b3, 5→b2 (13), 3→b3 (11), 7→b4: boxes (9,4), (6,2,5), (8,3), (7) – 4 boxes. (b) Decreasing order 9, 8, 7, 6, 5, 4, 3, 2: 9→b1, 8→b2, 7→b3, 6→b1 (15), 5→b2 (13), 4→b3 (11), 3→b4 (3), 2→b4 (5): boxes (9,6), (8,5), (7,4), (3,2) – 4 boxes. (c) Total weight $9+4+6+2+8+5+3+7 = 44$, and $\frac{44}{15} = 2.93\ldots$ so at least 3 boxes; both algorithms use 4 – first-fit decreasing is not guaranteed to beat first-fit, though it usually does.

3.2 (a) Pivot (position 4 of 7) = 4: less in order = 1, 3, 2; greater = 6, 8, 9, giving 1, 3, 2, **4**, 6, 8, 9. Sublist 1, 3, 2 (pivot 3): less = 1, 2; greater = none → 1, 2, **3**; then (1, 2) pivot 1: **1**, 2. Sublist 6, 8, 9 (pivot 8): less = 6; greater = 9 → 6, **8**, 9. Final: 1, 2, 3, 4, 6, 8, 9. (b) On large lists quick sort is generally much faster (fewer comparisons on average), because each partition splits the problem instead of moving items one position per comparison.

Local review guidance: pivots stated and partitions listed; placements shown item by item with running totals; lower-bound comparison quoted. This is authored classroom guidance.