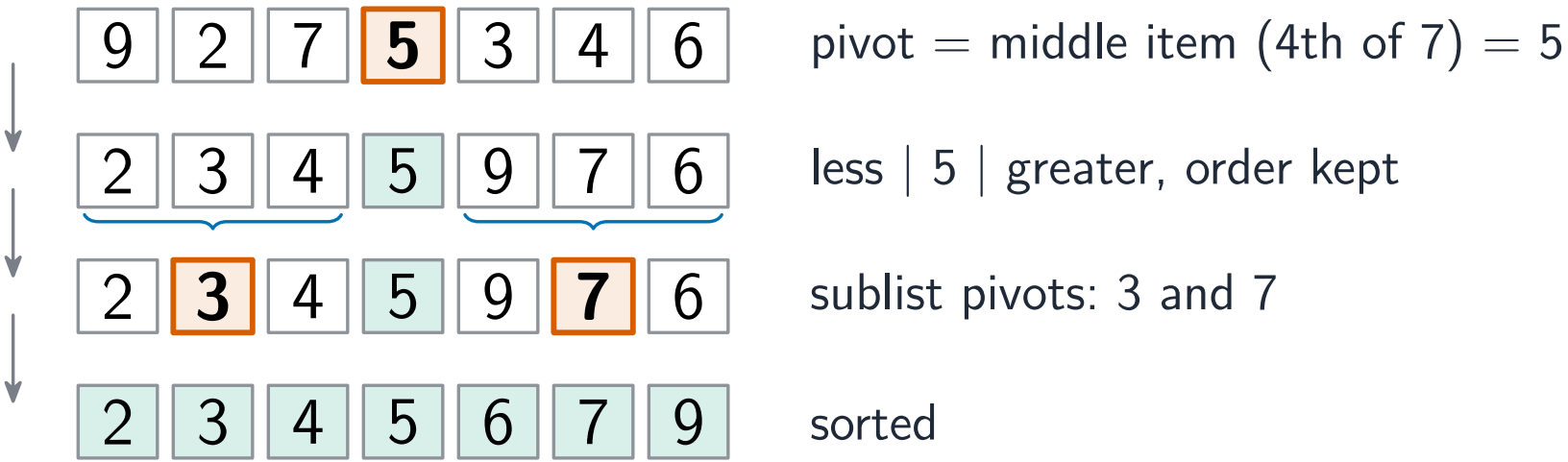
Exercise walk-through

10.1
Sorting and packing algorithms

Pearson Edexcel · International A-Level · Further Mathematics

Figure: Sorting and packing algorithms

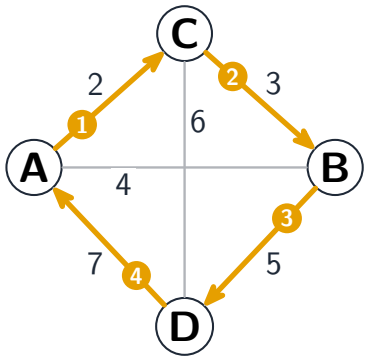


- **Quick sort** with the middle-item pivot: quote the pivot and the list after every partition
- Match each part of the diagram to the text.

The travelling salesman problem

- **Upper bound**: an actual route. Doubling the weight of the MST gives one; shortcuts improve it. The **nearest neighbour algorithm** also gives one: from the start, go to the nearest unvisited vertex each time, then return to the start.
- **Lower bound**: delete one vertex, find the MST of the rest, and add the two smallest edges from the deleted vertex.
- Nearest neighbour from A: A–C (2), C–B (3), B–D (5), D–A (7). Total 17, an upper bound.

Figure: The travelling salesman problem



nearest neighbour from A:
A–C–B–D–A $= 2+3+5+7 = 17$
upper bound

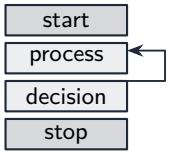
delete A: MST of B, C, D
$= 3+5 = 8$
plus AC + AB $= 2+4$:
lower bound 14

$14 \leq$ optimal tour ≤ 17

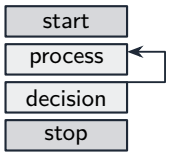
- Nearest neighbour gives an upper bound, the deleted-vertex MST a lower bound: the optimal tour lies between 14 and 17
- Match each part of the diagram to the text.

Exercises 2.1·2.2

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

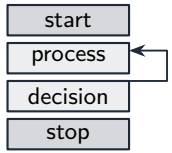


2.2 Quick sort 1, 6, 3, 5, 7, 2, 9, stating each pivot and the list after each partition. [4]



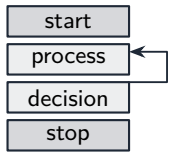
Exercises 2.1·2.2

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



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

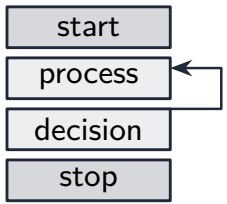


2.2 Pivot (position 4 of 7) = 5: less = 1, 3, 2; greater = 6, 7, 9, giving 1, 3, 2, **5**, 6, 7, 9. Sublist 1, 3, 2 (pivot 3): items less than 3, in order, are 1, 2 and none are greater, so the sublist becomes 1, 2, **3**; the pair 1, 2 (pivot 1) gives **1**, 2. Sublist 6, 7, 9 (pivot 7): less = 6, greater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

exercises.lol exercises.com.cn

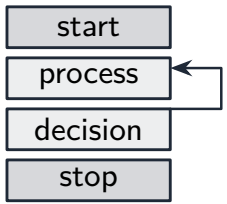
Exercise 2.3

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]



Exercise 2.3

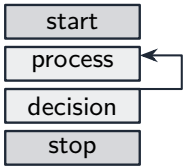
2.3 Bin capacity 8; items 5, 3, 4, 2, 6, 1. Apply first-fit, showing each placement, and state the number of bins used. [3]



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.

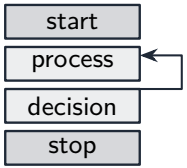
Exercise 3.1

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]



Exercise 3.1

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]

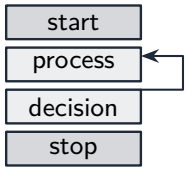


3.1 (a) First-fit (capacity 15), each parcel into the first box with room: 9→b1; 4→b1 (13); 6→b2; 2→b1 (15, full); 8→b2 (14); 5→b3; 3→b3 (8); 7→b3 (15). Boxes (9, 4, 2), (6, 8), (5, 3, 7) – 3 boxes. (b) Decreasing order 9, 8, 7, 6, 5, 4, 3, 2: 9→b1; 8→b2; 7→b2 (15, full; it does not fit b1, $9+7=16$); 6→b1 (15, full); 5→b3; 4→b3 (9); 3→b3 (12); 2→b3 (14). Boxes (9, 6), (8, 7), (5, 4, 3, 2) – 3 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 are needed; both algorithms use exactly 3, so both packings are optimal here (first-fit decreasing is not guaranteed to do better than first-fit, and in this list it does no better).

Exercise 3.2

3.2 *(Original.)* The list 1, 6, 3, 4, 8, 2, 9 is to be sorted into ascending order.

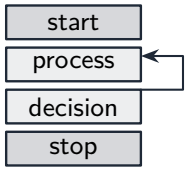
(a) Use the quick sort algorithm, stating the pivot at each stage and the list after each partition. (b) State one advantage quick sort has over bubble sort on large lists. [8]



Exercise 3.2

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]



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.