Solutions

Marking: **M1** a correct method, **A1** the correct answer, **B1** an independent mark.

Each answer shows the steps, then the **result**.

2.1

`[3, 7, 5, 1]`

`[3, 5, 7, 1]`

`[1, 3, 5, 7]`

2.2

$\log_2 16 = \mathbf{4}$ levels

each level makes fewer than n comparisons: at most $16 \times 4 = \mathbf{64}$

2.3

the values below 4 move to the front in order: 3, 2, 1

`[3, 2, 1, 4, 8, 5]`

the pivot is at index 3

2.4

a stable sort keeps names in order within each grade: Ben, Dara, then Ana, Chen

merge sort (or insertion or counting sort)

3.1

(a)

```
def merge(a, b):
    out, i, j = [], 0, 0
    while i < len(a) and j < len(b):
        if a[i] <= b[j]:
            out.append(a[i]); i += 1
        else:
            out.append(b[j]); j += 1
    return out + a[i:] + b[j:]
```

compare the two fronts; `<=` keeps it stable; copy whatever is left

(b) $T(n) = 2T(n/2) + cn$; $T(n) = \Theta(n \log n)$

(c) $O(n)$ for the merged output

(d) $\log_2 1024 = \mathbf{10}$

3.2

(a) each partition puts every other item on one side, so the parts have sizes $n-1$, $n-2$, …; comparisons $n(n-1)/2 = \mathbf{499\,500}$

exercises.lol exercises.com.cn

(b) no input order is bad for every random choice; the expected time is $O(n \log n)$ for every input

(c) median of three: the middle value of the first, middle and last items

3.3

(a) count how many records have each score, in an array of 101 counters; turn the counts into start positions and copy each record to its place

(b) about $n + k \approx 10^8$ steps; against $10^8 \times 26.6 \approx 2.7 \times 10^9$: about 27 times fewer

(c) the bound applies only to sorts that compare items; counting sort uses the key as an array index, and works only for small integer keys

(d) stable counting sort, filling the output from the start positions in input order; input and output take $2 \times 10^8 \times 16$ B $= 3.2$ GB, which fits in 8 GB