Solutions

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

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

2.1

in-order: 1 3 6 8 10 14

pre-order: 8 3 1 6 10 14

post-order: 1 6 3 14 10 8

2.2

each key is the right child of the one before: a chain of height 6

a search can take n steps: $O(n)$

2.3

parent $(4-1)//2 = \mathbf{1}$

children $2 \times 4 + 1 = \mathbf{9}$ and $\mathbf{10}$

2.4

remove 1; move the last value, 3, to the root: `[3, 4, 2, 7, 5]`

3 is larger than its smaller child 2: swap, giving `[2, 4, 3, 7, 5]`

3 at index 2 has no children, so stop; 1 was returned

3.1

(a) 45 root; 20 and 60 below it; 10, 30 under 20; 50, 70 under 60; 25 is the left child of 30

(b) the longest path is 45, 20, 30, 25: height **3**

(c) 20 has two children, so it is replaced by its in-order successor, the smallest key on its right: **25**

(d) 10 25 30 45 50 60 70

3.2

(a) index 2: 7 swaps with 2, giving `[9, 4, 2, 1, 8, 7]`; index 1: 4 swaps with 1, giving `[9, 1, 2, 4, 8, 7]`; index 0: 9 swaps with 1, then with 4: `[1, 4, 2, 9, 8, 7]`

(b) most nodes are near the bottom and sift down only 0 or 1 levels; the total work is a sum like $n/4 \cdot 1 + n/8 \cdot 2 + \cdots < n$, while each of n inserts may sift up $\log n$ levels

3.3

(a) heap: add and take are both $O(\log n)$, about $\log_2 10^4 \approx 14$ swaps; sorted list: take from the end is $O(1)$, but an add shifts about $n/2 = 5000$ items

(b) a heap only keeps parents $\leq$ children, so equal keys come out in no fixed order; store `(priority, arrival_number)`, so ties go by arrival

exercises.lol exercises.com.cn

(c) keep an index from patient to heap position and sift the entry up (decrease-key); or push a new entry and mark the old one as cancelled, skipping it when it comes out (lazy deletion)

(d) yes: sorting a copy of 10^4 entries once a minute takes about $10^4 \times 14$ steps, which is tiny, while every add and take stays $O(\log n)$