Solutions

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

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

2.1

$4T(n/2) + n$: $n^{\log_2 4} = n^2$ beats n: $\Theta(n^2)$

$2T(n/2) + n^2$: n^2 beats n: $\Theta(n^2)$

$3T(n/3) + n$: equal: $\Theta(n \log n)$

2.2

$T(n) = n$: $\Theta(n)$

halving: $\Theta(\log n)$

2.3

levels: n, $3n/4$, $9n/16$

a geometric series with ratio $3/4 < 1$

total $\le 4n$: $\Theta(n)$

2.4

bit i flips every 2^i increments: $n + n/2 + n/4 + \dots$

$< 2n$, so $O(1)$ amortised per increment

3.1

(a) $T(n) = 3T(n/2) + \Theta(n)$; $\Theta(n^{\log_2 3}) = \Theta(n^{1.585})$

(b) $\Theta(n^{\log_2 4}) = \Theta(n^2)$

(c) $3^{10} = \mathbf{59\,049}$; $4^{10} = \mathbf{1\,048\,576}$; about **17.8** times fewer

(d) Karatsuba has more additions and overhead, so it is slower until n is fairly large

3.2

(a) $T(n) = 7T(n/2) + \Theta(n^2)$; $\Theta(n^{\log_2 7}) \approx \Theta(n^{2.81})$

(b) $(8/7)^{12} \approx \mathbf{5.0}$

(c) more rounding error and more memory, and the many additions cost more for small blocks

3.3

(a) fill to exactly n (full), then push: it doubles, copying n; pop: it is half full and shrinks, copying again; repeat push, pop: every operation copies about n

(b) after any resize the array is half full; at least $n/4$ cheap operations must happen before the next resize

(c) each resize copies at most about as many elements as the operations since the last resize, so each operation pays $O(1)$ toward copying: $O(m)$ in total

(d) double when full and halve at a quarter full: $O(1)$ amortised per operation with at most 4 times the space actually used; for a hard limit, reserve 10 000 slots once

exercises.lol exercises.com.cn