Solutions

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

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

2.1

base $n = 1$: $1 = 1^2$

assume $1 + \cdots + (2k-1) = k^2$; add $2k+1$

$k^2 + 2k + 1 = (k+1)^2$, so the statement holds for $k+1$

2.2

calls `fact(4)`, `fact(3)`, `fact(2)`, `fact(1)`, `fact(0)`

returns 1, 1, 2, 6, **24**

2.3

$a_2 = 5$, $a_3 = 9$, $a_4 = 17$

$a_n = 2^n + 1$

$2^4 + 1 = 17$

2.4

it prints 3, 2, 1, 0, -1, … until a `RecursionError`: there is no base case

add `if n < 0: return` (or `if n == 0: return`) before the recursive call

3.1

(a) $4 > 4$, $8 > 9$, $16 > 16$ are all false; the statement first holds at $n = 5$, so the base case is 5

(b) $2^5 = 32 > 25 = 5^2$

(c) assume $2^k > k^2$ for some $k \geq 5$; then $2^{k+1} = 2 \times 2^k > 2k^2$; $2k^2 = k^2 + k^2 \geq k^2 + 2k + 1 = (k+1)^2$; so $2^{k+1} > (k+1)^2$, and the claim holds for all $n \geq 5$

3.2

(a) **1024**

(b) 10, 5, 2, 1, 0: **5** calls

(c) each call halves n, so there are about $\log_2 n$ calls, each with at most two multiplications, against $n - 1$ for the loop

3.3

(a) **8**

(b) $3, 5;\ 9, 15$

(c) base: $C(0) = 1 = 2F(1) - 1$ and $C(1) = 1 = 2F(2) - 1$; assume true for $n-1$ and $n-2$ (strong induction); $C(n) = 1 + 2F(n) - 1 + 2F(n-1) - 1 = 2F(n+1) - 1$

exercises.lol exercises.com.cn

(d) the remembering version: the naive one repeats the same sub-calls and grows like $F(n)$, so it is about 87,000 times slower at $n = 30$ and soon unusable