Solutions

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

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

2.1

offset $\log_2 32 = \mathbf{5}$

lines $65\,536/32 = 2048$, index **11**

tag $32 - 11 - 5 = \mathbf{16}$

2.2

$0 \bmod 4 = 4 \bmod 4 = 0$: both use line 0

so all **4** are misses: each evicts the other

2-way: both fit in set 0, so only the first 2 miss

2.3

$2 + 0.04 \times 50$

$= \mathbf{4}$ ns

2.4

temporal: `total` and `i`, used in every pass

spatial: the elements of `a`, read one after another

3.1

(a) blocks $32\,768/64 = 512$; sets $512/4 = \mathbf{128}$; offset 6, index 7; tag $32 - 7 - 6 = \mathbf{19}$

(b) offset: the low 6 bits, `0x38` = **56**; index: the next 7 bits, **89**

(c) a block can go in any of 4 places in its set, so two busy blocks with the same index no longer evict each other; but a hit must compare 4 tags and choose between 4 blocks, which adds delay

3.2

(a) L2 term: $10 + 0.2 \times 100 = 30$ ns; $1 + 0.05 \times 30 = \mathbf{2.5}$ ns

(b) $1 + 0.05 \times 100 = \mathbf{6}$ ns

(c) $6/2.5 = \mathbf{2.4}$

3.3

(a) a block holds $64/8 = 8$ numbers, so 1 miss in 8; $2^{20}/8 = \mathbf{131\,072}$ misses

(b) each step jumps 1024×8 bytes to a new block, which has been evicted by the next pass; every access misses: **1 048 576**

(c) $(1\,048\,576 - 131\,072) \times 100$ ns $\approx$ **91.8 ms**

exercises.lol exercises.com.cn

(d) blocking (tiling): work on small square tiles that fit in the cache, so each block loaded is used many times before it is evicted; or store a transposed copy so both reads are row by row