Tutorial 4：KV Cache 与状态组织

题目

1. [单选题] KV Cache 更接近下列哪一类系统对象？A. 只在单个函数里临时存在的局部变量 B. 跨阶段持续驻留的状态对象 C. 只用于前端展示的文本对象 D. 与调度完全无关的装饰信息

---

2. [判断题] 只要两个请求前缀相同，就可以直接认定系统已经获得真实复用收益。

---

3. [多选题] 下列哪些现象更像是 KV 压力上升的表现？A. 排队时间变长 B. 并发能力下降 C. 长上下文持续挤压其他请求 D. 所有请求的计算路径都变成常数时间

---

4. [简答题] 为什么长上下文不应只被理解为“计算量更大”？

5. [计算题] 若每个 block 可容纳 16 个 token，一个请求当前已有 50 个 token。请计算它至少需要多少个 block，以及最后一个 block 中有多少个 token。

---

6. [判断题] 只要 block 设计得足够大，复用率就一定更高，且浪费更少。

---

7. [简答题] 为什么 KV 问题通常首先表现为容量问题？

8. [伪代码题] 请写出一个简短伪代码：若 prefix_hash_hit 为真则尝试复用 block；否则从空闲列表中分配新 block。

9. [多选题] 若要证明“前缀复用真正带来了收益”，至少还需要哪些证据？A. 命中率 B. 维护与回收成本 C. 端到端时延或吞吐变化 D. 只展示某次命中截图即可

---

10. [简答题] 如何从外部症状判断问题更像 KV 管理，而不是纯算力不足？

参考答案

1. B。KV Cache 属于跨阶段持续驻留的状态对象。
2. 错。还需要证明命中、维护和回收成本都足够划算。
3. A、B、C。这些都是较典型的 KV 压力信号。
4. 因为它还会带来更长的状态驻留时间和更大的容量占用。
5. 至少需要 4 个 block；最后一个 block 中有 2 个 token。
6. 错。块太大容易浪费，块太小则会抬高管理成本。
7. 因为问题首先体现为“还能容纳多少请求”，而不是“单次是否能计算完成”。
8. 示例伪代码：

```
function get_block(prefix_hash_hit, free_blocks):
     if prefix_hash_hit:
          return reuse_block()
     return allocate(free_blocks.pop())
```

9. A、B、C。仅有命中现象本身并不足以证明真实收益。
10. 当等待变长、并发下降且长上下文持续挤压其他请求时，更像是 KV 管理问题。