Tutorial 2：请求生命周期与 Prefill/Decode

题目

1. [单选题] 若用户长期等待首句输出，应优先怀疑生命周期中的哪一段？A. 排队与 prefill B. decode 稳态 C. 结果打印格式 D. 日志写盘

---

2. [判断题] 如果 TPOT 基本稳定，就可以断定 prefill 一定没有问题。

---

3. [简答题] 为什么只画“请求进入”和“结果返回”两点，通常不足以支撑系统分析？

4. [排序题] 请按常见顺序排列以下阶段：回收、排队、decode、prefill、请求接入。

---

5. [计算题] 某请求在 0.0s 到达，1.8s 输出首 token，4.8s 完成，共输出 12 个 token。请计算 TTFT 与平均 TPOT。

---

6. [多选题] 下列哪些现象更像排队问题，而不是 decode 问题？A. TTFT 明显上升 B. TPOT 基本不变 C. 首 token 明显变慢 D. 每个输出 token 的间隔持续抖动

---

7. [伪代码题] 请写出一个简短伪代码，根据 `has_first_token` 和 `is_finished` 两个状态，判断请求当前处于 prefill、decode 还是 finished。

8. [判断题] 生命周期图中的资源回收点只用于画图，对性能解释没有作用。

---

9. [简答题] 为什么 TTFT 与 TPOT 会自然绑定到不同阶段?

10. [多选题] 下列哪些对象通常会跨阶段持续存在? A. 规范化输入状态 B. Sequence 状态 C. KV 状态 D. 一次性的标题文本

参考答案

1. A。首句延迟首先表现为排队和 prefill 一侧的问题。
2. 错。TPOT 稳定只能说明后半程较稳，不能自动排除 prefill 问题。
3. 因为这种画法无法说明排队、prefill、decode 和回收等关键边界。
4. 请求接入 -> 排队 -> prefill -> decode -> 回收。
5. TTFT = 1.8s；TPOT = (4.8 - 1.8) / (12 - 1) 约等于 0.27s。
6. A、B、C。TTFT 上升而 TPOT 近似不变时，更像是排队或前半程问题。
7. 示例伪代码：

```
function stage(has_first_token, is_finished):
     if is_finished:
          return "finished"
     if has_first_token:
          return "decode"
     return "prefill"
```

8. 错。没有回收点，就难以解释资源何时释放以及何处形成回压。
9. 因为 TTFT 更受前半程影响，而 TPOT 更受 decode 稳态影响。
10. A、B、C。这些对象通常都会跨阶段存在。