

任务描述：

创建一个模拟厨房订单执行情况的实时系统。
厨房每秒应收到 2 个订单。厨房应在收到订单后立即烹饪(假设烹饪时间为 0)，然后将订单放在最可用的餐架上（参见餐架部分）等待服务员取餐。
收到订单后，系统应派遣服务员来取餐并派送该订单。服务员应在 2-6 秒后随机到达。服务员应在抵达后立即取餐,取餐后服务员应立即派送(假设送餐时间为 0)。
请优先使用 golang 语言完成，其他语言 python,java,c 等也可。

订单

测试订单文件使用附件中的 orders.json。订单必须从文件中解析出来，并以每秒 2 个订单的速度输入到模拟系统中。订单输入速度需要可以配置，方便面试官测试模拟系统的行为。每个订单只能输入一次，当所有订单都被处理完毕时，模拟系统应该停止。

```
{
  "id": "2ec069e3-576f-48eb-869f-74a540ef840c",
  "name": "Acai Bowl",
  "temp": "cold", // 订单适宜的温度
  "shelfLife": 249, // 订单餐架寿命，单位为秒
  "decayRate": 0.3 // 订单衰减率
},
```

餐架

厨房取餐区有多个餐架，可存放不同温度的餐饮。每个订单都应放在与订单温度相匹配的架子上。如果该餐架已满，则可以放置在溢出餐架。如果溢出餐架已满，则应从溢出餐架已有订单中选择一个移动到温度适宜的且有剩余空间的餐架上。如果无法移动，则随机从溢出餐架上已有订单中丢弃一个（丢弃后不能被服务员取餐）。

下表详细介绍了厨房的架子：

名称	适宜温度	容量
Hot Shelf	Hot	10
Cold Shelf	Cold	10
Frozen Shelf	Frozen	10
Overflow shelf	Any temperature	15

餐架寿命

订单的价值会随着时间的推移而恶化，具体取决于订单的餐架寿命和衰减率字段。价值小于等于零的订单不应该取餐和送餐，应该从餐架上移除。在系统输出中显示订单时，请显示当前订单值。

订单价值公式

$$\text{value} = \frac{(\text{shelfLife} - \text{orderAge} - \text{decayRate} * \text{orderAge} * \text{shelfDecayModifier})}{\text{shelfLife}}$$

注：

orderAge 表示订单下单到现在的时长(单位为秒)

shelfDecayModifier 表示衰减率调整系数，溢出餐架为 2，其余餐架为 1

评分标准

- ☐ 满足题目中描述的所有规定要求
- ☐ 是有效的可运行代码
- ☐ 具有控制台或日志输出，允许面试官在运行时清楚地了解和跟踪系统的操作。每当系统中发生事件（订单接收、提取、丢弃等）时，系统应输出一条消息，其中包含触发事件的描述和餐架内容的完整列表
- ☐ 有一个 README 文件，包括如何在本地环境运行和测试代码，以及订单移入和移出餐架的核心算法和逻辑