

编程目标

请实现一个调度服务器，提供webservice，接受客户端的任务请求。

客户端任务请求的格式如下，是一组正整数序列

2, 4, 100, 6, 10, 90, 1, 1, 10, 2, 15, 30, 1, 5, 9, 10

该序列表示一组任务的执行时间片，每个数字代表一个任务完成所需的时间，都是整数。

示例任务列表如下，其中任务索引由调度服务器维护。

任务索引 (task_index)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
任务时间片 (秒)	2	4	100	6	10	90	1	1	10	2	15	30	1	5	9	10

任务 0 需要 2 个时间单位才能完成，任务 1 需要 4 个时间单位，以此类推。

假设系统的调度带宽为 5，也就是说在每个时间单位（每秒）内，调度器最多只能安排总共 **5个时间片** 的任务来执行。

请实现一个调度服务器，提供webservice，接受客户端的任务请求。程序运行后不断输出每秒的调度状态，包括：

- 当前时间点 (time)
- 当前调度的任务索引 (index)
- 被调度任务的剩余执行时间 (remain_time_slice)

示例输出（前几轮）：

时间 time	执行任务索引 index	剩余执行时间片 remain_time_slice
0	0, 1	0, 1（任务 0: 2-2=0，任务 1: 4-3=1）
1	1, 2	0, 96（任务 1: 1-1=0，任务 2: 100-4=96）
2	2	91（任务 2: 96-5=91）

要求程序支持以下两种调度策略

策略一：顺序调度（FIFO）

规则说明：

- 按照任务在列表中的顺序进行调度
- 每次调度的任务之和不得超过带宽限制（5 个时间单位）；
- 同一个任务可以跨多个时间周期执行（若未完成则继续排队）
- 当某个任务剩余时间为 0 时视为完成

策略二：短任务优先（SRTF，Shortest Remaining Time First）

规则说明：

- 在所有可调度的任务中，优先选择剩余执行时间越短的任务
- 如果剩余时间相同，优先调度索引编号更小的任务
- 每次调度的任务之和不得超过带宽限制（5 个时间单位）
- 一旦任务被调度，其剩余时间将被扣减
- 任务未完成将返回队列，根据实时任务优先级，重新调度

针对上述两种策略，请实现程序使其能够完整输出所有时间点上的调度情况，并支持选择策略。

注意事项

- 请使用Golang完成本题
- 请考虑客户端的并发请求的情况
- 请添加相关测试