# 边缘集群AI推理的分布式任务调度

> 时间限制：30s 空间限制：512MB

## 背景信息

在多用户共享算力资源的场景下，由于推理任务的启动时间不同步、性能需求差异显著，加之服务器本身的异构性，系统会面临复杂的调度需求。

需要选手设计调度算法，尽可能满足多个用户的吞吐量需求。

## 题目描述

多个用户在不同时刻有不同的推理请求，为处理请求提供多种服务器。选手需要通过把请求放置于服务器上，并采用迁移操作使得尽可能满足吞吐量需求。

以下是相关信息的详细描述：

- 服务器：提供$N$台服务器，第$i$台服务器有$g_i$个$NPU$，显存大小是$m_i\ MB$。单个服务器的所有$NPU$是相同的。
- 推理耗时：第$i$种服务器的单个$NPU$同时处理多个请求时，该$NPU$上的第$j$个请求的耗时是$\lceil \frac{B_j}{f(B_j)} \rceil$毫秒，其中$f(B_j) = k_i * \sqrt{B_j}$表示推理速度，$k_i$表示该$NPU$的推理速度系数，$B_j$表示第$j$个请求的$batchsize$。请求在第$x$毫秒开始推理，推理耗时$y$毫秒，则该请求在第$x + y$毫秒完成推理。
- 显存：$Memory = a * batchsize + b$表示显存与$batchsize$的关系。$b$是模型相关固有参数。
- 模型：每个请求使用的模型都相同，每个$NPU$都已预加载该模型。
- 用户：共有$M$个用户有推理诉求。其中第$i$个用户希望在第$[s_i, e_i)$毫秒内推理$cnt_i$个样本。
- 通信时延：第$i$个用户把请求发送到第$j$种服务器的通信时延是$latency_{j,i}$毫秒，即用户在第$x$毫秒发送请求，服务器在第$x + latency_{j,i}$毫秒收到请求。用户可在第$x + latency_{j,i} + 1$毫秒发送下一个请求。
- 迁移次数：形式化地，第$i$个用户在同一时刻只能发送一个请求，记接收该请求的$NPU$编号是$v$。按照时间顺序，该用户发送第一个请求到最后一个请求形成的$v$序列是$V$。记$V$的长度是$L$，$move_i = \sum_{i=1}^{L-1}[V_i! = V_{i+1}]$，则该用户得分随$move_i$增加而减少，具体细节见评分规则。
- 推理顺序：每个$NPU$对应一个队列，队列内存储**未完成推理**的请求。队列排序的第一关键字是请求到达该$NPU$所在服务器的时刻，越早到达的请求越靠近队首。对于同时到达的请求，编号越小的用户发送的请求越靠近队首。每毫秒会处理一次队列。按照先后顺序执行的处理方式如下：

1. 移除已完成推理的请求。
2. 增加当前时刻接收到的请求。
3. 将序列排序。
4. 从队首至队尾依次扫描请求，若加上该请求所需显存后未超过服务器显存，则认为本毫秒对该请求分配推理资源。

备注：本题的最小时间单位是毫秒，不可拆分。

# 题目交互

## 输入

第一行一个整数$N(1 \leq N \leq 10)$表示服务器种类数。

接着$N$行，每行3个整数$g_i, k_i, m_i(1 \leq g_i \leq 10, 1 \leq k_i \leq 5, 1000 \leq m_i \leq 2000)$，其中$g_i$表示第$i$台服务器$NPU$个数，$k_i$表示推理速度参数，$m_i$表示$NPU$显存大小。与上文“服务器”含义一致。

接着一行，该行一个整数$M(1 \leq M \leq 500)$表示用户数量。

接着$M$行，每行3个整数$s_i, e_i, cnt_i(0 \leq s_i < e_i \leq 60000, 1 \leq cnt_i \leq 6000, 5 \times cnt_i \leq e_i - s_i)$，其中$s_i, e_i$表示用户的推理请求时间段$[s_i, e_i)$，$cnt_i$表示待推理样本数量。

接着$N$行，每行$M$个整数，其中第$i$行第$j$个整数$latency_{i,j}(10 \leq latency_{i,j} \leq 20)$表示第$j$个用户把请求发送到第$i$种服务器的通信时延。

接着一行，该行两个整数$a, b(10 \leq a \leq 20, 100 \leq b \leq 200)$表示显存和$batchsize$的关系。

## 输出

输出$2M$行，第$2i-1$行和第$2i$行表示第$i$个用户的推理方案。

第$2i-1$行包含一个整数$T_i$，需要保证$1 \leq T_i \leq 300$。

第$2i$行包含$4T_i$个整数，第$4j-3$个整数$time_j(0 \leq time_j < time_{j+1} \leq 1000000, time_{T_i+1} = +\infty)$，第$4j-2$个整数$server_j(1 \leq server_j \leq N)$，第$4j-1$个整数$NPU_j(1 \leq NPU_j \leq g_i)$，第$4j$个整数$B_j(1 \leq B_j \leq 1000), a \times B_j + b \leq m_{server_j}$，表示第$i$个用户在$time_j$时刻将包含$B_j$个推理样本的请求发送给第$server_j$台服务器的第$NPU_j$个$NPU$。选手需保证$time_1 \geq s_i, \sum_{j=1}^{T_i} B_j = cnt_i$。

# 评分规则

单个测试用例得分$Score = h(K) \times \sum_{i=1}^{M} h(\frac{end_i - e_i}{e_i - s_i}) \times p(move_i) \times 10000$，其中$end_i$表示该用户的最后一个样本推理完成的时刻，$K = \sum_{i=1}^{M} [end_i > e_i]$，$h(x) = 2^{-\frac{x}{100}}$，$p(x) = 2^{-\frac{x}{200}}$。

选手总得分为所有测试用例得分之和。

# 样例

## 输入

```
2

2 3 10000

6 5 20000

3

0 60000 3000

10000 50000 2000

0 200000 5000

10 20 20

20 10 12

10 100
```

## 输出

```
1

0 1 1 3000

2
```

> 0 1 1 1000 20000 1 2 1000
>
> 2
>
> 0 2 1 1000 100000 1 2 2000

备注：输入输出仅为说明格式。

# 错误类型

## 基础错误类型

1. 代码编译错误
2. 程序异常退出 (可能原因: 运行错误，使用异常权限，输出参数比实际多，输出参数格式不对，etc...)
3. 超出时间限制 (可能原因: 交互时未使用清空流缓存命令，程序运行超时，输出参数比实际少，etc...)
4. 超出内存限制

## 逻辑错误类型

- "Unknown Error" (出现时请联系大赛方)
- "RE" (程序内存访问越界或异常退出)
- "Invalid Output"（$T_i$或者$time_j$违反约束）
- "Batchsize Exceeds Memory"（$B_j$违反约束）
- "Samples Not Fully Processed"（样本未全部发送）
- "Invalid Time Order"（$time_j$非递增）
- "Invalid NPU Index"（NPU编号违反约束）
- "Invalid User Send Time"（样本发送时刻早于允许发送的时刻）
- "Invalid Server Index"（服务器编号违反约束）