



第七次作业指导书



第一部分：训练目标



本次作业的基本目标是模拟**多线程实时电梯系统**，在前两次作业的基础上，掌握线程之间的交互，强化线程之间的协同设计层次架构



第二部分：基本概念

- 1、电梯系统时间 T_{real} ：从程序开始运行，到所有线程终止，程序结束的时刻花费的时间，即程序的真实运行时间 (real time)
- 2、电梯系统运行花费总时间 T_{final} ：从程序开始运行，到电梯最后输出关门的时刻花费的时间
- 3、开关门窗口时间：从电梯开始开门的时刻到完成关门的时刻的时间闭区间
- 4、数据基本限制：所有测试数据均需满足的限制
- 5、公测数据限制：中测和强测数据满足的限制
- 6、互测数据限制：互测数据需满足的限制

第三部分：题目描述

一. 电梯系统说明

本次作业需要模拟一个多线程实时电梯系统，该电梯系统支持动态扩展和日常维护，同时需要支持更加高级的调度功能。

系统基于一个类似北京航空航天大学新主楼的大楼，电梯可以在楼座内 1 — 11 层之间运行。

系统从标准输入中输入请求信息，程序进行接收和处理，模拟电梯运行，将必要的运行信息通过输出接口进行输出。

具体而言，本次作业电梯系统具有的功能为：上下行，开关门，模拟乘客的进出，以及**模拟电梯系统扩建和日常维护时乘客的调度**。

电梯系统**可以采用任意的调度策略**，即在任意时刻，系统选择上下行动，是否在某层开





< 第七次作业 - 2023面向对象设计与构造



电梯系统默认初始在 1 层有六部电梯，这些电梯均可达所有楼层。



二. 电梯系统参数说明



1. 初始电梯参数



- 1、可到达楼层：1 - 11 层
- 2、初始位置：1 层
- 3、数量：6 部
- 4、编号：6 部电梯，ID 分别为 1 — 6
- 5、移动一层花费的时间：0.4s
- 6、开门花费的时间：0.2s
- 7、关门花费的时间：0.2s
- 8、限乘人数：6 人



2. 电梯系统调度参数

- 对任意楼层 X ，处于服务中的电梯的最大电梯数量 $M_X = 4$
- 对任意楼层 X ，处于服务中的只接人的电梯的最大数量 $N_X = 2$

两个参数的解释见第六部分——公测说明。

三. 电梯请求说明

在电梯的每个入口，都有一个输入装置，让每个乘客输入自己的目的位置。电梯基于这样的一个目的地选择系统进行调度，将乘客运送到指定的目标位置。

所以，一个电梯请求包含这个人的**出发楼层和目的楼层**，以及这个人的 id（**保证人员 id 唯一**），请求内容将作为一个整体送入电梯系统，在整个运行过程中请求内容不会发生改变。

支持电梯动态添加功能，具体而言，添加电梯请求中包含**新电梯的ID和初始楼层**，程序需要将这部电梯加入到电梯系统的调度中。而在本次作业中，需要额外支持**电梯定制化**功能，电梯相关参数将由指令给出。

支持电梯日常维护功能，具体而言，日常维护请求包含**需要维护的电梯ID**，则这部电梯需要尽快（具体评判标准见公测说明部分）将电梯内所有乘客放出然后退出电梯调度系统，程序还需要进行必要的重新调度来满足这些乘客的请求。

本次作业需要支持电梯的可达性，即一部电梯不一定能在所有楼层停靠（但显然可以经过所有楼层）。可达性将以掩码形式给出。





< 第七次作业 - 2023面向对象设计与构造



在这个以性能规划为前提下，我们对电梯的捎带策略不做任何限制。希望同学们多多阅读并自行探索相关算法，对电梯的捎带策略有自己的设计与思考。

为保证同学们实现的都为作业要求的**可捎带电梯**，我们对电梯的性能做一定的约束，采用**ALS调度策略**为性能**基准**（关于调度策略，欢迎大家积极上网找资料探索）。性能约束详见公测说明-正确性判断章节中关于 T_{max} 的部分。

五.基准策略

对于每一个电梯，都采用 ALS 策略，即新增**主请求**和**被捎带请求**两个概念

- 1、主请求选择规则：
 - (1)、如果电梯中没有乘客，将请求队列中到达时间最早的请求作为主请求
 - (2)、如果电梯中有乘客，将其中到达时间最早的乘客请求作为主请求
- 2、被捎带请求选择规则：
 - (1)、电梯的主请求存在
 - (2)、该请求投喂的时刻**小于等于**电梯到达该请求出发楼层关门的截止时间
 - (3)、电梯的运行方向和该请求的目标方向一致
- 3、在电梯的调度上，采用一种较为均衡的调度方式，例如 7 层有5个乘客，那么第1个乘客分配给第1部电梯，第2个乘客分配给第2部电梯，第3个乘客分配给第3部电梯，第4个乘客分配给第1部电梯，第5个乘客分配给第2部电梯。
- 4、在电梯维护上，接收到维护指令的电梯会在接收到 MAINTAIN 指令后停靠在最近的楼层并立刻放出所有乘客。
- 5、对无法使用一部电梯完成的请求采用静态调度策略，即事先规划好完成该请求时需要乘坐的电梯顺序，然后按照该顺序乘坐电梯，当这些电梯中有的电梯在还未乘上便进入维护时则重新规划。标程在规划时会选择乘坐电梯数最少的方案。

第四部分：输入/输出说明

一.输入输出接口说明

- 1、本单元作业使用官方提供的输入输出接口进行输入输出。
- 2、输入接口提供了若干方法，通过调用方法可以直接获取所需数据对象。输入接口在**尝试**读取标准输入的内容时，对于正确的输入将成功解析，错误的则输出错误信



- 4、详细的接口定义和使用方法见官方接口说明文档。
- 5、程序的输入输出为**实时交互**，评测机可以做到在某个时间点投放一定量的输入。

二.输入数据

- 1. 乘客请求
 - (1).格式为： [时间戳]乘客ID-FROM-起点层-TO-终点层
 - (2).输入保证起点层和终点层不同。
 - (3).乘客 ID 唯一
- 2. 增加电梯请求
 - (1).格式为：
[时间戳]ADD-Elevator-电梯ID-起始楼层-满载人数-移动一层的时间-可达性
 - (2).输入保证起始楼层在整栋楼的范围以内。
 - (3).电梯ID 唯一，且保证在程序逻辑正确的前提下该电梯ID不存在于当前电梯系统中。
- 3. 日常维护请求
 - (1).格式为： [时间戳]MAINTAIN-Elevator-电梯ID
 - (2).输入保证在程序逻辑正确的前提下，电梯ID对应的电梯已存在于当前电梯系统中
 - (3).若被维护的电梯为新增电梯，保证维护指令在该电梯新增指令的2s或以后才输入

三.输出数据

- 1. 通过调用输出接口的每一次输出将自动附加时间戳于头部，具体请参考**输出接口文档**，请不要试图伪造时间戳，否则会在实际评测时产生不一样的结果导致程序运行错误。
- 2. 电梯到达某一位置： [时间戳]ARRIVE-所在层-电梯ID
- 3. 电梯开始开门： [时间戳]OPEN-所在层-电梯ID
- 4. 电梯完成关门： [时间戳]CLOSE-所在层-电梯ID
- 5. 乘客进入电梯： [时间戳]IN-乘客ID-所在层-电梯ID
- 6. 乘客离开电梯： [时间戳]OUT-乘客ID-所在层-电梯ID
- 7. 电梯接收到日常维护请求： [时间戳]MAINTAIN_ACCEPT-电梯ID （该消息由官方包自动输出，同学无需输出）
- 8. 电梯可以开始进行日常维护： [时间戳]MAINTAIN_ABLE-电梯ID





[1.7]ADD-Elevator-7-5-5-0.6-178

[2.1]MAINTAIN-Elevator-1

[5.6]2-FROM-5-TO-1

[8.6]MAINTAIN-Elevator-7

[9.6]1-FROM-4-TO-5

[2.2810]MAINTAIN_ACCEPT-1

[2.2820]MAINTAIN_ABLE-1

[6.1960]ARRIVE-2-2

[6.6020]ARRIVE-3-2

[7.0060]ARRIVE-4-2

[7.4220]ARRIVE-5-2

[7.4220]OPEN-5-2

[7.6260]IN-2-5-2

[7.8330]CLOSE-5-2

[8.2340]ARRIVE-4-2

[8.6410]ARRIVE-3-2

[8.7820]MAINTAIN_ACCEPT-7

[8.7820]MAINTAIN_ABLE-7

[9.0480]ARRIVE-2-2

[9.4550]ARRIVE-1-2

[9.4560]OPEN-1-2

[9.4560]OUT-2-1-2

[9.8670]CLOSE-1-2

[10.2720]ARRIVE-2-2

[10.6750]ARRIVE-3-2

[11.0820]ARRIVE-4-2

[11.0820]OPEN-4-2

[11.2840]IN-1-4-2

[11.4850]CLOSE-4-2

[11.8920]ARRIVE-5-2

[11.8920]OPEN-5-2

[11.8920]OUT-1-5-2

[12.2970]CLOSE-5-2

178=000_1011_0010b, 所以新加的7号电梯可到达2, 5, 6, 8层

[1.4]MAINTAIN-Elevator-1

[1.9]ADD-Elevator-7-1-7-0.4-1545

[4.1]1-FROM-2-TO-9

[1.5820]MAINTAIN_ACCEPT-1

[1.5830]MAINTAIN_ABLE-1

[4.6780]ARRIVE-2-2

[4.6790]OPEN-2-2

[4.8920]IN-1-2-2





< 第七次作业 - 2023面向对象设计与构造



[8.9]MAINTAIN-
Elevator-7

[
5.4820]MAINTAIN_ACCEPT-
2
[5.4830]OPEN-3-2
[5.4830]OUT-1-3-2
[5.8840]CLOSE-3-2
[5.8840]MAINTAIN_ABLE-2
[5.8990]ARRIVE-2-3
[6.3040]ARRIVE-3-3
[6.3040]OPEN-3-3
[6.5070]IN-1-3-3
[6.7090]CLOSE-3-3
[7.1140]ARRIVE-4-3
[7.5180]ARRIVE-5-3
[7.9250]ARRIVE-6-3
[8.3320]ARRIVE-7-3
[8.7390]ARRIVE-8-3
[
9.0810]MAINTAIN_ACCEPT-
7
[9.0810]MAINTAIN_ABLE-7
[9.1440]ARRIVE-9-3
[9.1440]OPEN-9-3
[9.1440]OUT-1-9-3
[9.5490]CLOSE-9-3

[1.9]2-FROM-10-
TO-6
[5.6]3-FROM-11-
TO-6
[6.6]1-FROM-3-
TO-5
[8.5]MAINTAIN-
Elevator-1
[10.2]ADD-
Elevator-7-5-6-
0.2-437

[2.4850]ARRIVE-2-1
[2.8920]ARRIVE-3-1
[3.2950]ARRIVE-4-1
[3.7000]ARRIVE-5-1
[4.1060]ARRIVE-6-1
[4.5070]ARRIVE-7-1
[4.9080]ARRIVE-8-1
[5.3130]ARRIVE-9-1
[5.7180]ARRIVE-10-1
[5.7180]OPEN-10-1
[5.9200]IN-2-10-1
[6.1250]CLOSE-10-1
[6.1880]ARRIVE-2-2
[6.5320]ARRIVE-9-1



< 第七次作业 - 2023面向对象设计与构造



[6.9990]ARRIVE-4-2
[7.1860]ARRIVE-2-3
[7.3390]ARRIVE-7-1
[7.4020]ARRIVE-5-2
[7.6020]ARRIVE-3-3
[7.6020]OPEN-3-3
[7.7530]ARRIVE-6-1
[7.7530]OPEN-6-1
[7.7540]OUT-2-6-1
[7.8140]ARRIVE-6-2
[7.8140]IN-1-3-3
[8.0170]CLOSE-3-3
[8.1570]CLOSE-6-1
[8.2200]ARRIVE-7-2
[8.4210]ARRIVE-4-3
[8.6240]ARRIVE-8-2
[
8.6710]MAINTAIN_ACCEPT-
1
[8.6720]MAINTAIN_ABLE-1
[8.8260]ARRIVE-5-3
[8.8260]OPEN-5-3
[8.8260]OUT-1-5-3
[9.0290]ARRIVE-9-2
[9.2300]CLOSE-5-3
[9.4330]ARRIVE-10-2
[9.8360]ARRIVE-11-2
[9.8360]OPEN-11-2
[10.0380]IN-3-11-2
[10.2420]CLOSE-11-2
[10.6470]ARRIVE-10-2
[11.0550]ARRIVE-9-2
[11.4610]ARRIVE-8-2
[11.8620]ARRIVE-7-2
[12.2670]ARRIVE-6-2
[12.2680]OPEN-6-2
[12.2680]OUT-3-6-2
[12.6690]CLOSE-6-2

[1.4]ADD-

[2.6730]OPEN-1-1





< 第七次作业 - 2023面向对象设计与构造



[2.5]2-FROM-1-TO-6
[2.7]1-FROM-10-TO-9
[2.9]ADD-Elevator-8-1-6-0.2-645
[8.0]MAINTAIN-Elevator-1
[8.8]MAINTAIN-Elevator-7

[3.2800]ARRIVE-2-2
[3.5000]ARRIVE-2-1
[3.6890]ARRIVE-3-2
[3.9070]ARRIVE-3-1
[4.0920]ARRIVE-4-2
[4.3090]ARRIVE-4-1
[4.4980]ARRIVE-5-2
[4.7180]ARRIVE-5-1
[4.9070]ARRIVE-6-2
[5.1260]ARRIVE-6-1
[5.1270]OPEN-6-1
[5.1270]OUT-2-6-1
[5.3130]ARRIVE-7-2
[5.5310]CLOSE-6-1
[5.7180]ARRIVE-8-2
[6.1230]ARRIVE-9-2
[6.5290]ARRIVE-10-2
[6.5300]OPEN-10-2
[6.7350]IN-1-10-2
[6.9380]CLOSE-10-2
[7.3460]ARRIVE-9-2
[7.3460]OPEN-9-2
[7.3460]OUT-1-9-2
[7.7580]CLOSE-9-2
[8.1810]MAINTAIN_ACCEPT-1
[8.1820]MAINTAIN_ABLE-1
[8.9760]MAINTAIN_ACCEPT-7
[8.9770]MAINTAIN_ABLE-7

[1.7]ADD-Elevator-7-5-5-0.2-178
[9.6]1-FROM-5-TO-8

[9.8840]OPEN-5-7
[10.0980]IN-1-5-7
[10.3000]CLOSE-5-7
[10.5020]ARRIVE-6-7
[10.7060]ARRIVE-7-7
[10.9080]ARRIVE-8-7
[10.9090]OPEN-8-7

电梯正常情况下只能在可达楼层开门，但是经过的楼层无论可达不可达均需要输出ARRIVE信息。





< 第七次作业 - 2023面向对象设计与构造



- 1、为了万便说明，指导书上提供的输入为这样的格式：[x.x]yyyyyyyyy。
- 2、意思是在 x.x 这个时刻（相对于程序运行开始的时间，单位秒），输入 yyyyyyyyyy 这样的一行数据。
- 3、关于上面说到的**时间计测同步性**问题，在这里解释一下：
 - (1)、直观地说，可能会观察到输入指令时间晚于做出反应的时间或反应时间明显晚于预期时间的现象。
 - (2)、这个实际上不是程序的错误，而是由于评测机投放数据的系统、和程序输出接口，这两边的时间可能存在不同步导致的。
 - (3)、**这一问题是由于多线程测试不可避免的波动性导致的计时同步性误差，属于正常现象，不会被认定为错误。**（当然，也不会出现故意提早的情况，因为程序本身就是实时交互，不可能做得到预知未来）但是，电梯的操作还是应当遵从逻辑，不能出现乘客先进电梯，再开门这样的操作，你的程序所输出的操作的时间戳也应当是递增的。

第五部分：限制说明

一.数据基本限制

- 可输入电梯系统的指令数：1~100条（指令数为0或超过100均为不合法输入）。
- 系统时间上限 T_{max} ：电梯系统时间 T_{real} 和电梯系统运行花费总时间 T_{final} 的上限时间，**一旦超过将直接判定为TIME_LIMIT_EXCEED**。 T_{max} 与标程运行时间 T_{std} 的计算公式在下文给出。
- 乘客ID保证不重复，存在于电梯系统内的电梯ID保证不重复，所有ID均为int范围内正整数。
- 乘客请求中保证起点层和终点层不同
- 输入数据的时间戳精确到**小数点后一位**
- 输入数据中保证存在至少一条乘客请求
- 新增加的电梯需满足以下限制
 - 移动一层的时间 $\in \{0.2, 0.3, 0.4, 0.5, 0.6\}$
 - 满载人数 $\in \{3, 4, 5, 6, 7, 8\}$
 - 至少能到达三个或以上的楼层
- ADD 指令添加的电梯ID保证不重复
- MAINTAIN 指令中的电梯ID必须存在于指令输入时的电梯系统中。
- 任意时刻，电梯系统保证能完成任意请求，即对于任意请求，总能找出一个楼层序列，序列开头为起点层，结尾为终点层，乘客能够通过使用电梯系统依次经过序列中各楼层直到到达终点层。





- 公测中 T_{max} 与标程运行时间 T_{std} 的计算公式如下。

$$T_{max} = \max(T_{std} + 10, 1.15 \times T_{std})$$



- 包括强测在内的数据都会保证正常程序的电梯系统时间不会超过 T_{max} ，也会避免使用对边界情况较为敏感的数据。
- 对于中测和强测数据， T_{max} 将会严格限制为通过上述公式计算得到的值。所以，请不要试图用超长的sleep来回避线程安全停止的问题。
- 任意时刻，电梯系统中的电梯数量不超过12

三.互测数据限制

- 系统时间上限 T_{max} ：对于互测数据， T_{max} 为220s。
- 第一条指令的投喂时间在1s或1s以后。
- 最后一条指令的输入时间不晚于50s。
- 指令条数不超过70。
- 随测试点提交的输出合法
- 初始六台电梯中至多有四台电梯被维护
- 任意时刻，电梯系统中的电梯数不超过10

第六部分：公测说明

一. 正确性判断

- 1、**代码实现中不存在轮询**。轮询是一种非常占用CPU资源的行为，为了避免出现轮询的情况，**请使用Wait-Notify的方式编程**。同时，为了检查是否出现轮询的情况，总CPU时间限制为10s，不满足该限制的程序会被判定为错误。
- 2、**电梯的性能符合基本要求**，即你的程序需要满足 $\max(T_{real}, T_{final}) \leq T_{max}$ 。
- 3、**输出结果符合逻辑**，具体来说，有以下几点：
 - (1)、电梯运行逻辑合理：
 - a、电梯到达一层后就可以输出开门信息，然后花0.2s开门，开门结束。
 - b、电梯开门结束后，某一刻电梯花0.2s关门，然后输出关门信息，就可以离开楼层。

电梯在该楼层间若发生移动，应满足移动时间要求





< 第七次作业 - 2023面向对象设计与构造



- (2)、人员进出逻辑合理：
 - a、只有已经在电梯里的人可以出电梯，也只有不在电梯里的人可以进入电梯。
 - b、如果电梯在楼层 2，那么乘客显然也只能在楼层 2 进出。
 - c、**反常识设定**：开门的 0.2s 期间或关门的 0.2s 期间乘客仍然可以进出，即在**整个开关门窗口时间内**乘客都可以进出。纸片人没有厚度
 - d、乘客进出信息在电梯开关门窗口时间之内。

```
1 | 错误示例：
2 | [ 4.6280]OUT-1-1 ← 这个人没开门就出来了
3 | [ 4.8440]OPEN-1
4 | [ 5.0470]CLOSE-1
```

- e、电梯系统结束运行时，所有的乘客都到达了目标楼层，且没有被困在电梯里。
- (3)、电梯载客量合理：
 - a、**电梯任何时候，内部的人数都必须小于等于轿厢容量限制。**
 - b、也就是说，即便在 OPEN、CLOSE 中间，也不允许出现超过容量限制的中间情况。
- (4)、日常维护逻辑合理：
 - a、未接收到 MAINTAIN 指令的电梯不得进行维护
 - b、收到 MAINTAIN 指令的电梯必须在**两次移动楼层操作内**将所有乘客放出并输出“电梯可以开始日常维护”部分的输出。
 - “在两次移动楼层操作内”指的是在官方包自动输出 MAINTAIN_ACCEPT 开始，该部电梯在输出 MAINTAIN_ABLE 之前至多输出两条 ARRIVE 。
 - c、电梯输出 MAINTAIN_ABLE 时必须保证电梯轿厢内没有人，且门是关着的。
 - d、输出 MAINTAIN_ABLE 后的电梯不得参与后续的电梯系统调度
 - e、**收到日常维护指令的电梯不受其自身可达性影响**，可停靠在任何楼层。
 - 因为可达性实际上是通过我们程序去约束其不可达而非电梯硬件自身去实现不可达性。当遇到突发状况时由梯系统会解除对官方电梯的可达性限制。五重安全





< 第七次作业 - 2023面向对象设计与构造



在某一时刻 t ，某一楼层 X ，对电梯行为给出如下定义

- 1、**未服务**：电梯不处于 X 层，或者电梯处于 X 层但未开门。
- 2、**服务中**：电梯处于 X 层且电梯处于开门状态。

每一次开关门会对应着一次电梯内外乘客的交换，针对某部电梯一次开关门时间窗口内的乘客交换，对该电梯的行为给出如下定义。

- 1、**只接人**：设该电梯开门前电梯内乘客集合为 $before$ ，关门后电梯内乘客集合为 $after$ ，则二者满足以下子集关系： $before \subseteq after$ 。

则一个程序被认为是电梯调度逻辑合理的，当且仅当

- 1、在任意时刻，任意楼层 X ，处于服务中的电梯数量小于等于 M_X 部
- 2、在任意时刻，任意楼层 X ，处于服务中的只接人电梯数量小于等于 N_X 部
- (6)、时间戳合理：即**输出的时间戳应该是非减的**。

二.性能分的评判

在强测时，性能分将由三部分组成：系统运行时间，等待时间与期望时间之差的最大值和系统耗电量。其中，系统运行时间即为 $T_{run} = \max T_{real}, T_{final}$ 。最大等待时间与期望时间之差计算方法如下。

对第 i 个请求，设其等待时间为 $T_i = T_{\text{到达目标楼层}} - T_{\text{发出请求}}$ ，该请求的期望等待时间 ET_i 计算公式如下。

$$ET_i = \frac{\sum_{f=F_{min}}^{F_{max}} |F_s - f| T_{move}}{F_{max} - F_{min} + 1} + T_{open} + T_{close} + |F_d - F_s| T_{move}$$

F_{min} ：电梯系统最低楼层

F_{max} ：电梯系统最高楼层

F_s ：请求的起始楼层

F_d ：请求的目标楼层

T_{open} ：开门时间

T_{close} ：关门时间

T_{move} ：电梯移动时间

本次作业，上述公式中为定值的参数如下。

$$F_{min} = 1$$

$$F_{max} = 11$$



< 第七次作业 - 2023面向对象设计与构造



整个公式中，后三个数是电梯将该乘客送往目的楼层的所需时间，第一个项是在假设电梯在各楼层的概率分布为均匀分布时电梯到达起始楼层的期望。则等待时间与期望时间之差的最大值为

$$MT = \max_i (T_i - ET_i)$$

系统耗电量将使用以下方法计算。

根据输出信息中的 ARRIVE, OPEN, CLOSE 信息的数量进行计算。

- 每条 ARRIVE 信息表示电梯移动一层，耗电量 $W_{ARRIVE} = 0.4$
- 每条 OPEN(CLOSE) 信息表示电梯开门（关门）一次，耗电量 $W_{OPEN} = W_{CLOSE} = 0.1$
- 系统的耗电量 $W = W_{OPEN}N_{OPEN} + W_{CLOSE}N_{CLOSE} + W_{ARRIVE}N_{ARRIVE}$ ，其中 $N_{OPEN}, N_{CLOSE}, N_{ARRIVE}$ 分别表示输出中 OPEN, CLOSE, ARRIVE 信息的总数。

设对于所有同学的数据，参数 x 的平均值为 x_{avg} ，最大值为 x_{max} ，最小值为 x_{min} ，则定义以下折算函数

$$\begin{aligned} base_{min} &= p \cdot x_{avg} + (1 - p) \cdot x_{min} \\ base_{max} &= p \cdot x_{avg} + (1 - p) \cdot x_{max} \\ p &= 0.25 \\ r(x) &= 100\% \cdot \begin{cases} 1 & x \leq base_{min} \\ 1 - 10^{1 - \frac{base_{max} - base_{min}}{x - base_{min}}} & base_{min} < x \leq base_{max} \\ 0 & x > base_{max} \end{cases} \end{aligned}$$

最终性能分 s 将由以下公式计算。

$$s = 15 \times (0.3r(T_{run}) + 0.3r(MT) + 0.4r(W))$$

其中 T_{run}, MT, W 为你的程序的运行时间，等待时间与期望时间之差的最大值和系统耗电量。

第七部分：提示与警示

提示

- 1、关于接口的一些使用，可以在 idea 里面，按 Ctrl+Q。将可以看到类和方法的使用格式以及注释（本次官方提供的接口均提供了中文版 javadoc 注释，Ctrl+Q 可以直



```
1  import com.oocourse.elevator3.TimableOutput;
2
3  public class MainClass {
4      public static void main(String[] args) {
5          TimableOutput.initStartTimestamp(); // 初始化时间戳
6          /*
7              代码部分
8          */
9      }
10 }
```

- 3、官方提供的输出包是**线程安全**的。
- 4、由于**多线程调度存在随机性**，在使用官方提供的投喂脚本或进行互测环节时，可能会出现**输出与官方评测机不一致、bug 无法复现的情况**，最终结果均以**官方评测机为准**。
- 5、性能公式可能比较复杂，但实际上公式表达出的意思就是希望你的程序在**保证正确性的前提下**能够尽量做到以下三点。
 - 运行时间尽量短
 - 尽量不让请求等待过长时间
 - 尽量减少系统的无效运行

掩码

本次作业中的可到达信息 M 主要采用了掩码 (mask) 来表示，下面进行一个简单的举例：

如果我们要表示物品是易燃还是易爆，那么可以采取如下方式：

- 我们用二进制第 1 位（即最低位）的 0 / 1 来表示 不易燃 / 易燃
- 我们用二进制第 2 位的 0 / 1 来表示 不易爆 / 易爆

这样的话，我们只需要两位就能表示出所有的情况：

- 0 (二进制 00) 代表不易燃且不易爆
- 1 (二进制 01) 代表易燃但是不易爆
- 2 (二进制 10) 代表易爆但是不易燃
- 3 (二进制 11) 代表易燃且易爆

需要得到这个物品的情况时就取出其对应位即可。



< 第七次作业 - 2023面向对象设计与构造



$$access \& (1 \ll (floor - 1)) \begin{cases} \neq 0 & \text{可达} \\ = 0 & \text{不可达} \end{cases}$$

(*access*的二进制表示的第*i*位对应第*i* + 1层的可达性)

警示

不要试图Hack评测机，不要抄袭。如发现其他人的代码疑似存在上述行为，可向课程组举报。课程组感谢同学们为课程建设所作出的贡献。

