1 摘要

本文通过对排队问题的随机模拟，研究随机顾客输入流下的不同办事大厅设定的服务情况，同时对服务情况进行一些指标的统计，同时对模拟的不足提出一种可能的实际解决方案。

2 实验目的

2.1 问题描述

设有一个系统，比如银行，顾客们到达并排队等候，直到在总计 k 个服务台钟有一个是空闲的。顾客的到达情况由概率分布函数控制，服务时间（当服务台有空时，用于服务的时间量）也是如此。我们关心的是平均一位顾客要等多久，或所排的队伍可能有多长这类统计问题。

当顾客输入流大于服务台的个数的时候，就形成了排队现象。顾客到达办事大厅进行排队等待直到 k 个服务台中有至少一个更新为空闲状态。我们关注在给定的办事大厅设定下（例如服务总时长与服务台个数），对顾客输入进行模拟，并且得到对以下指标的统计：

1. 顾客平均等待时长
2. 最大等待人数
3. 办事大厅关闭时仍未被服务的人数
4. 对顾客考虑忍耐阈值的属性，并统计因无法忍耐而离开的人数。

2.2 解决方案

对一整个办事大厅的服务的模拟，可以归结为每一个时间点（tick）上可能发生的事情的模拟，而在每一个 tick 上可能发生的事情主要取决与当前办事大厅的状态，而服务的实行者是服务台，服务的主体是顾客，因而在每一个 tick 里面，都要查询服务台的状态，并且查询等待顾客的状态，对于空闲的服务台，则查询有没有正在等待的顾客，若有，则更新服务台的信息；若没有空闲的服务台，那

么意味着已经到达的顾客需要继续等待，那么对当前正在等待的顾客进行等待时间的累加。

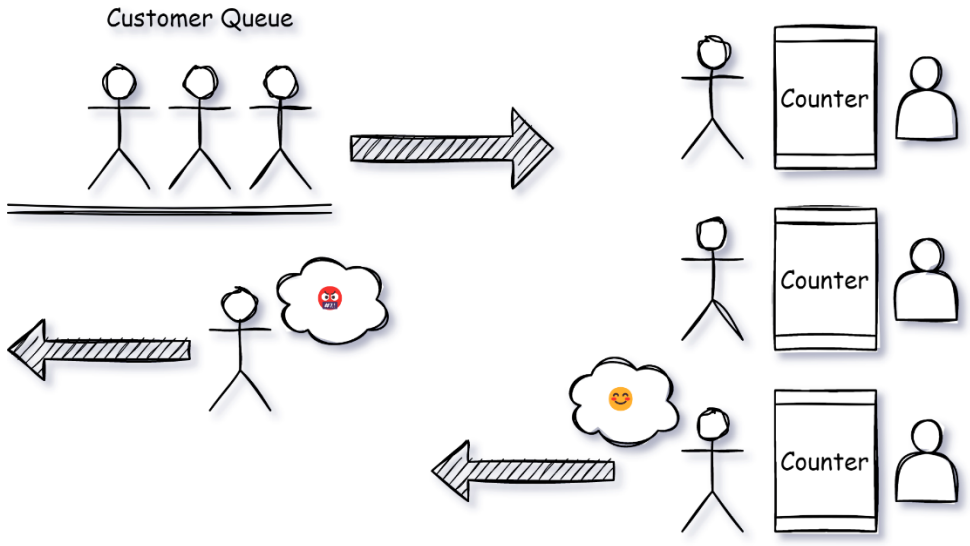

Figure 1 **运营情况模拟图**

2.3 达成的目的

统计得到顾客的平均等待时长，整个办事大厅的最大等待人数，办事大厅关闭时仍未被服务的人数，以及因无法忍耐而离开的人数。

3 实验环境与技术工具

环境名称	具体环境
操作系统	Windows10
处理器	Intel i5-1135G7 4 核
软件工具	Vs code+MinGW

表格 1 实验环境与软件

4 建立问题的数学模型

模型主体大概分为两个部分，第一个是对具体对象的抽象，第二个是对事件

的模拟。

具体实际对象主要有顾客，顾客队列，服务台。在这里我们关注这些对象的内在属性，对于顾客，可以建立顾客类，关注顾客的到达时间，服务时间，容忍上限；对于顾客队列，实际上这是一个按照顾客到达时间排序的数组；对于服务台，则需要关注服务剩余时间，为了方便输出顾客信息，保留一个属性用于记录当前的服务顾客号码是合理的，而服务台的空闲状态判断显然可以借用服务剩余时间来完成，因此不需要额外的变量来存储。

而对具体对象的抽象的过程中，最重要的是对类属性的抽象。本文采用随机数的方法模拟顾客的到达时间、服务时间与容忍上限。

对事件的模拟，以服务台为主体，对于正在工作的服务台，如果此刻没有工作完成，则通过修改剩余服务时间来表述“使其继续工作”，对于完成服务而进入空闲状态的服务台，则需要查询当前是否有正在等待的顾客，若有则需要接入服务并且更新信息，同时使得被服务顾客从输入流中出列。

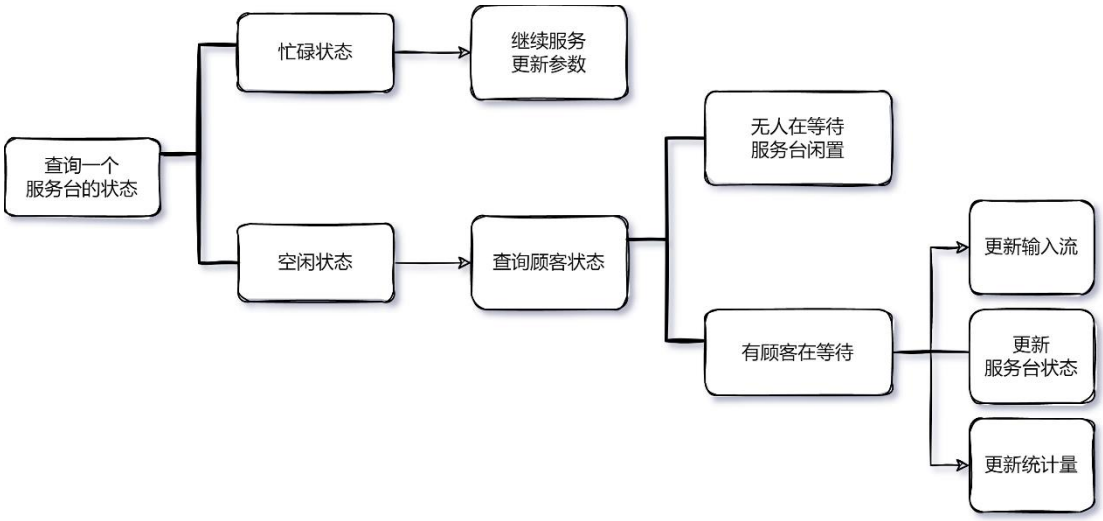

Figure-2 算法大致思路图

5 模型中的数据结构

本文中的数据结构主要应用于对顾客输入流的描述，并且只用 vector 实现即可。

由于不同顾客的到达时间不尽相同，并且应当按照到达时间的将顾客从输入流中选出，这容易让人想到优先队列，并且以顾客到达时间为关键字，然而在优先队列的数据结构中，我们只关注队首，队内的顾客情况对我们来说是不便知道的，当我们想要查询当前等待人数的时候，势必会有对队内情况的查询，而单一的优先队列在此类的队内顾客情况上查询上会受到阻碍。

因而，本文提出用 vector 来描述顾客输入流，为了在操作上与优先队列达到相似，有必要先对顾客进行关于到达时间的排序，而为了避免时间复杂度上的增大，本文中的 vector 对顾客输入流进行从大到小的排序，而不是从小到大。理由在于，顾客服务完成后势必要出列，若仅是采用“指针移动到下一位顾客”来表达当前顾客出列就会造成运行期间的空间浪费，若采用从小到大，那么 vector 删除队首元素毫无疑问是费时的，因而对顾客输入流采用从大到小排序，并且用 pop_back()函数来描述顾客出列的行为是更优的选择。

6 模型细节解释

假设一天办事大厅开放 10 个小时，本文中的 ticks 采用秒作为单位，因而有 36000 个 ticks，顾客的到达时间则呈现随机的概率分布。以下是对不同指标在本次实验中的处理方式。

顾客平均等待时长。在每一个 ticks 中，当一个顾客被服务时，意味着当前顾客的等待状态终止，因而只需要将当前顾客的到达时间与 ticks 作差即可知道顾客的等待时长，对所有顾客的等待时长进行累加，在办事大厅结束服务后取平均即可。

最大等待人数。在每一轮对服务台状态的查询完毕并且分配好应当被服务的顾客之后，对顾客队列，也就是 vector 中的末尾元素进行遍历，若被遍历到的顾客已经到达，那么显然应该被计数，当遍历到第一个没有到达的顾客之后，即可结束便利。这样就可以得到每一个时刻的等待人数，从而通过比较记录下最大的等待人数。

办事大厅关闭时仍未被服务的人数。当 ticks 遍历完毕后，仍然留在顾客队列中的顾客显然就是未被服务的，因而借助 vector 中的 size()函数可以容易的得

到对未被服务的人数的统计。

因为等待时间过长而离开的人数。用随机法生成顾客的容忍上限，在本研究中将顾客的容忍下限默认为该名顾客服务时间的 1/2，并利用随机数生成一个增量，该增量介于 0 与该名顾客服务时间的 1/3 之间，从而随机生成每一位顾客的容忍上限。在统计每一个 tick 的等待人数的时候，若发现顾客的等待时长已经超出容忍上限，那么进行计数，同时将顾客出队以表示这位顾客的离开。

为了便于观察每一个时刻的情况，提供一个程序运行过程中对情况观察的示例：

```
============================第 20510 秒 ============================
顾客到达情况：此时有 0 名顾客到达店铺
顾客服务情况：
Counter 1 : 正在服务顾客 154 , 剩余服务时间为 93
Counter 2 : 正在服务顾客 155 , 剩余服务时间为 249
Counter 3 : 正在服务顾客 156 , 剩余服务时间为 603
Counter 4 : 正在服务顾客 152 , 剩余服务时间为 96
今日剩余人数为：120        当前等待人数为：11
============================第 20511 秒 ============================
顾客到达情况：此时有 0 名顾客到达店铺
顾客服务情况：
Counter 1 : 正在服务顾客 154 , 剩余服务时间为 92
Counter 2 : 正在服务顾客 155 , 剩余服务时间为 248
Counter 3 : 正在服务顾客 156 , 剩余服务时间为 602
Counter 4 : 正在服务顾客 152 , 剩余服务时间为 95
一位顾客由于等待时间过长而离开.....
今日剩余人数为：119        当前等待人数为：9
```

同时也给出结果的运行示例：

```
=====================================================
平均等待时长为：3.54517 mins
服务结束后剩余人数为:0
最大等待人数为：11
由于等待时间过长而离开的人数为：44
```

7 实验结果

在本次实验中，分别以顾客数量为 200,250,300，服务台数量为 4,5，顾客服务时长上限为 600,900 作为对实际情况的模拟，并且记录下统计结果如下。其中平均等待时长以分钟为单位。

(C,K,L)	平均等待时长	服务结束后人数	最大等待人数	提前离开人数
(200,4,600)	0.43	0	4	0
(250,4,600)	0.96	0	5	1
(300,4,600)	1.28	0	8	8
(200,4,900)	1.25	0	6	5
(250,4,900)	2.16	0	8	16
(300,4,900)	3.56	0	11	44
(200,5,600)	0.12	0	3	0
(250,5,600)	0.26	0	4	0
(300,5,600)	0.58	0	5	0
(200,5,900)	0.59	0	5	0
(250,5,900)	1.18	0	6	0
(300,5,900)	2.15	0	9	8

表格 2 实验结果记录

根据上表，显然在在一些情况下出现服务台资源是过剩的，例如当顾客数量为 200，服务台数量为 3 个，顾客服务时长上限为 10 分钟时，顾客平均等待时长不到 1 分钟，甚至服务时长的 1/10，此时增加服务台数量则只会增大这种浪费。在顾客数量为 300，服务台数量为 4，顾客服务时长上限为 15 分钟时，等待人数峰值达到 11 位，同时因为无法忍受长时间等待而离开的人数高达 44 位，这在一定程度上反映了服务台资源的紧张；这种现象在增加一台服务台之后得到缓解，尽管等待人数的峰值在 9 个，但因为无法等待而离开的人数为 8 人，可以认为服务台资源紧张的情况得到缓解。

8 模型的不足与可能的改进

在上表(300,4,900)的办事大厅示例中，因为无法忍受而离开的人数高达 44 位，然而平均等待时长却不到 4 分钟，这意味着在绝大部分时候，服务台资源其实是处于相对空闲的状态，而当顾客集中进入办事大厅时又不可避免的造成资源紧张，

从而导致顾客流失。因而，在实际中最合理的做法应当是设定备用的服务台，在闲期，备用服务台不开放，使得已经开放的服务台能够最大限度的被利用；在忙期，备用服务台开放，以避免顾客流失。