数据结构与算法课程实验报告

实验名称：模拟银行

摘 要

本实验旨在设计出一种有效的算法，根据给出银行营业时间、窗口数量、顾客到达时间和服务时间间隔、最大容忍时间等变量模拟出所有顾客的到达和离开事件。并分别设计允许VIP插队和VIP限时排队的排队策略，综合考虑银行、普通顾客、VIP顾客三方的诉求，探讨最优排队机制。在算法设计方面，本实验明确了实验的假设和声明，针对本实验的 3 个问题，详细阐述了相应算法的实现思路和原理。在实验结果部分，本实验针对具体问题，结合实际情况运行程序进行测试，并对实验结果进行了深入分析，探讨出排队机制对总体不满意度的影响。

关键词：模拟，排队机制，总体不满意度

1.1 实验内容

实现一种模拟银行排队的算法，旨在根据给出的银行营业时间、窗口数、顾客到达时间间隔范围、顾客业务办理时间范围等参数进行模拟，计算出顾客到达和离开事件的发生时间、办理业务的窗口、等待时间等。并分别设计允许VIP插队和VIP限时排队的排队策略，综合考虑银行、普通顾客、VIP顾客三方的诉求，探讨最优排队机制。具体实验内容包括：

增设VIP窗口前后的窗口繁忙情况、VIP顾客平均等待时间等参数。

- 设计并实现允许VIP顾客在普通窗口插队时的模拟算法。
- 设计并实现VIP顾客限时等待，即在等待时间较长时，允许其在普通窗口插队时的模拟算法。
- 设计并实现综合考虑银行、普通顾客、VIP顾客三方诉求的排队机制评价标准，探讨最优排队机制。
- 对实验结果进行分析和总结，评估算法的有效性和适用性。

1.2 实验假设

1.2.1 文件结构

为避免在解决下一问题时改动到上一问题的代码，本实验针对不同的问题提交了 4 个文件夹，其功能如下表所示。

表 1 文件夹名称对应题号与功能

文件夹名称	题号	功能
bank_ori	第 1 题第一问	不含VIP窗口
bank_1_1	第 1 题第一问	含有VIP窗口
bank_1_2	第 1 题第二问	允许VIP插队
bank_2_3	第 2 题和第 3 题	VIP限时等待和最佳排序机制

1.2.2 算法假设和声明

1) 题目所给的框架代码中文件sim.h的第 222 行有bug，nexttime应该改e.GetTime()

```
if ( nexttime > simulationLength )
      // process events but don't generate any more
      continue;
```

图 1 sim.h 第 222-224 行（修改前）

修改前的代码对于当前到达事件在银行营业时间以内，但下一到达事件在营业时间以外的到达事件不予处理。以下图所示情况为例，在第 8 分钟本应该有 3 号顾客到达，但由于此时的nexttime为 12，超过了营业时间，因此程序没有输出其到达信息，也没有将其离开信息添加到优先级队列中。所以此处将代码修改为判断到达时间是否超过营业时间，即可输出正确的结果。

```
Enter the simulation time in minutes: 10
Enter the number of bank tellers: 2
Enter the range of arrival times in minutes: 4 4
Enter the range of service times in minutes: 9 9
Time:  0  arrival of customer 1
Time:  4  arrival of customer 2
Time:  9  departure of customer 1
       Teller 1  Wait 0  Service 9
Time: 13  departure of customer 2
       Teller 2  Wait 0  Service 9
```

图 2 修改前代码运行结果

2) 题目中约定“用户之间无法感知对方是处理何种业务，不知道其业务所需的具体时间和剩余时间”，那么无论是普通顾客还是VIP顾客，在到达时都无法确定选择哪个窗口的等待时间最短；对于VIP顾客，在可以插队的情况下，也无法确定去VIP窗口排队还是去普通窗口插队的等待时间最短。如果满足这一约定，那么顾客在选择窗口时只能随机选择，毫无疑问等待时间较长。

因此，在问题的求解过程中，忽略此条约定，假定银行在每名顾客到达时就已经知道其办理何种业务，并且银行通过长期的数据分析已经得出了办理每种业务所需时间，从而由银行指定每名用户排哪个窗口，以提高服务效率，降低顾客等待时间。

3) 从第 1 题第二问起，本人修改了输出格式，使顾客的到达和离开事件均显示为一行，从而使程序输出更清晰明了。

4) 在下文的表述和程序输出中，P代表普通顾客，V代表VIP顾客。

2 算法思路

2.1 第 1 题：VIP 窗口

2.1.1 第一问

本小题主要求解增设VIP窗口前后各窗口繁忙程度的变化情况。

根据题目中“将在去掉到达时间与服务时间随机化的情况下，对同学们的代码进行正确性验证”和表格中的顾客类型、顺序和数量，本题中对于增设VIP窗口前后，均约定窗口数量为 2，营业时间为 10 分钟，到达时间间隔为 4 分钟，服务时间为 9 分钟。其中，对于增设VIP窗口的情况，约定 1 号窗口为VIP窗口，2 号窗口为普通窗口，三名顾客的类型和顺序为P、P、V。

bank_ori文件夹中的代码是题目所给的框架代码，除修改上述bug外没有改动，其功能是模拟增设VIP窗口前的情况。

bank_1_1 文件夹中的代码在框架代码基础上增设了VIP窗口。对于普通顾客，一定排在 2 号窗口；对于VIP顾客，当且仅当 1 号窗口非空且 2 号窗口为空时排在 2 号窗口，否则排在 1 号窗口。

通过分析上述两种情况程序输出结果，可以得出增设VIP窗口前后各窗口繁忙程度的变化情况。

2.1.2 第二问

本小题在允许VIP顾客到普通窗口插队的情况下模拟出规定时间段内的顾

客到达和离开事件，并分析此时VIP和普通顾客平均等待的差异。

为更贴近在银行排队的实际情况，本小题增加了“当顾客到达后发现等待时间过长，会立刻选择离开”的设定。设定“等待时间过长”的标准是超过服务时间上限的TIMES倍，默认TIMES为2。但是已经排队的普通顾客因被VIP顾客插队而等待时间过长，不会离开。即顾客放弃排队，离开银行只会发生在顾客到达的同一时刻。

题目中约定“离开事件与到达时间同时发生时，先处理离开事件”，但由于有上述“顾客到达后发现等待时间太长，直接离开”的设定，如果仍遵循上述约定，就会出现还没有到达就已经离开的情况。因此为该约定增加一个例外情况“对于同一顾客的离开和到达事件同时发生时（即顾客直接离开的情况），先处理到达事件。”当然，上述例外情况在编程实现时并不需要特殊处理，这与数据结构有关：框架代码中的优先级队列遵循在时间相同的情况下，先进的先出；而对于同一顾客的到达事件，必定在离开事件之前加入队列。

约定VIP顾客插队时，最多只能插到该队伍最后一名VIP顾客之后。

顾客到达后窗口的选择是解决本问题的关键，下面对不同类型的顾客分别阐述：

- 对于普通顾客：根据各普通窗口结束服务的时间求出其等待时间，选择等待时间最短的窗口。
- 对于VIP顾客：先根据各VIP窗口结束服务的时间求出其在VIP窗口排队的等待时间，再根据各普通窗口的情况求出其在普通窗口插队所需的等待时间。选择上述两种情况等待时间最短的窗口。

2.2 第2题：VIP限时等待

本小题是在规定VIP顾客等待时间不超过最大容忍时间的情况下模拟出规定时间段内的顾客到达和离开事件，并分析此时VIP和普通顾客平均等待的差异。

约定两个事件不同时发生时，发生得早的在前；同时发生时，优先级为离开>到达且插队>到达不插队。

第1题第二问是VIP顾客可以在到达后直接在普通窗口插普通用户的队，也

就是说VIP顾客在到达银行时就已经知道是去VIP窗口排队还是去普通窗口插队能使其等待时间最短。这会导致VIP顾客愿意选择到普通窗口插队的方式降低自己的等待时间，其等待时间远低于普通顾客。

上述插队机制会极大地增加普通顾客的不满意度，因此第 2 题不再用上述机制，而是要让VIP顾客的等待时间会超过最大容忍时间时才允许其插队。这种插队机制，一方面使VIP顾客的平均排队时间在最大容忍时间附近，大概率不会等待过长时间；另一方面也会降低VIP顾客插队的人次，避免增加普通顾客的不满意度。

顾客到达后窗口的选择是解决本问题的关键，下面对不同类型的顾客分别阐述：

- 对于普通顾客，选择窗口的方法和上一问保持不变，仍然是选择等待时间最短的窗口。当然这并不一定是全局最优的窗口，因为在选定窗口开始排队后，可能会被后来的VIP顾客插队，导致在该窗口的等待时间反而高于其他窗口。

- 对于VIP顾客，先求出在不插队的情况下的等待时间，即排在任意窗口（含VIP窗口）的队尾需要多少等待时间。如果上述等待时间没有超过最大容忍时间，那么就选择该窗口，不插队；否则，遍历所有普通窗口，依次向前插队（不能超过最后一个VIP顾客，已经开始办业务的顾客也不能被插队），一直到等待时间低于最大容忍时间或不能插队的时候停止，统计插了多少名普通顾客的队，选取插队次数最少的窗口。

  之所以选择插队次数最少的窗口，是因为被插队的普通顾客数量越多，这次插队造成的影响就越差，普通顾客越不满意。需要注意的是，并不是每次插队都能确保VIP顾客的等待时间降低到最大容忍时间以内（例如前面已经有很多VIP顾客或正在办理业务的顾客剩余办理时间很长）。因此，本题设定如果在任何普通窗口插队都不能使等待时间降低到最大容忍时间以内，不插队，选择之前求出的等待时间最短的窗口。

题目中规定“等待时间满的VIP顾客，可以更换柜台服务”，但由于本实验假定银行在每名顾客到达时就已经知道其服务时间，因此对于VIP顾客，其在到

达时就一定能确定选择哪个窗口能使其等待时间最短，而不需要到了最大容忍时间才发现去另一个窗口排队更优，也就不存在更换柜台的情况。具体来讲，对VIP顾客而言，其在到达后发生的一切事件都对其选择窗口没有意义：后来的VIP顾客不可能插队到先来者之前，后来的普通顾客更不可能影响VIP顾客的等待时间。

2.3 第 3 题：最佳排队机制

对于普通用户来讲，被VIP插队的用户越多，越不满；对VIP顾客来讲，等待的时间越长，越不满；对银行来讲，有人排队但有越多窗口闲置时，越不满。因此，本小题是在综合考虑银行、普通顾客、VIP顾客三方诉求的情况下，探讨综合最优的排队策略。

具体实现上，本小题的代码是在第 2 题的基础上，增加了计算和输出上述三方以及总体的不满意度的模块，从而分析比较在特定情况下的VIP顾客的最大容忍时间、银行营业时间、窗口数量设置等变量对不满意度的影响。

为了更好地衡量排队机制的优劣，本实验设计了不满意度这一指标，其计算方法如下：

设总体不满意度为D，银行、普通顾客和VIP顾客的不满意度分别为D_1、D_2、D_3，总体不满意度的计算公式如下。

$$D = 0.5 \times D_1 + 0.2 \times D_2 + 0.3 \times D_3$$

其中，银行的不满意度主要考虑窗口空闲时间所占的比例。

$$D_1 = \frac{100 \times \text{窗口数量} - \text{所有窗口总服务时间百分比}}{100 \times \text{窗口数量}}$$

普通顾客的不满意度主要考虑平均等待时间和平均被插队次数。

$$D_2 = 0.5 \times D_{21} + 0.5 \times D_{22}$$

其中，D_{21}和D_{22}分别考虑普通顾客的平均等待时间和平均被插队次数。

$$D_{21} = \frac{\text{平均等待时间} - 2 \times waitHigh}{2 \times waitHigh}$$

$$D_{22} = \frac{\text{平均被插队人次} - 0.1}{0.1}$$

VIP顾客的不满意度主要考虑平均等待时间。

$$D_3 = \frac{\text{平均等待时间} - waitHigh}{waitHigh}$$

以上各项，若结果小于 0，则以 0 计算。总体不满意越小，则该排队机制越优。本实验使用控制变量法，分别改变VIP顾客的最大容忍时间、银行营业时间、VIP和普通窗口数量，观察并分析总体不满意度的大小。

3 实验结果

3.1 第 1 题：VIP 窗口

3.1.1 第一问

运行bank_ori中的代码，设置银行营业时间为 10 分钟，窗口数量为 2 个，顾客到达事件间隔为 4 分钟，顾客服务时间为 9 分钟，得到在不增设VIP窗口时的顾客到达和离开情况。

```
******** NO VIP TELLERS ********
Time:  0  arrival of customer 1
Time:  4  arrival of customer 2
Time:  8  arrival of customer 3
Time:  9  departure of customer 1
       Teller 1  Wait 0  Service 9
Time: 13  departure of customer 2
       Teller 2  Wait 0  Service 9
Time: 18  departure of customer 3
       Teller 1  Wait 1  Service 9

******** Simulation Summary ********
Simulation of 18 minutes
   No. of Customers:  3
   Average Customer Wait: 0 minutes
    Teller #1  % Working: 100
    Teller #2  % Working: 50
```

图 3 不增设 VIP 窗口的模拟结果

和普通窗口各

增设 1 个VIP

运行bank_1_1 中的代码，设置银行营业时间为 10 分钟，VIP和普通窗口各 1 个，顾客到达事件间隔为 4 分钟，顾客服务时间为 9 分钟。得到在增设 1 个VIP窗口时的顾客到达和离开情况。

```
******** ONE VIP TELLER ********
Time:  0  arrival of customer 1 isVip:0
Time:  4  arrival of customer 2 isVip:0
Time:  8  arrival of customer 3 isVip:1
Time:  9  departure of customer 1
       Teller 2  Wait 0  Service 9
Time: 17  departure of customer 3
       Teller 1  Wait 0  Service 9
Time: 18  departure of customer 2
       Teller 2  Wait 5  Service 9

******** Simulation Summary ********
Simulation of 18 minutes
   No. of Customers:  3
   Average Customer Wait: 2 minutes
    Teller #1  % Working: 50
    Teller #2  % Working: 100
```

图 4　增设 1 个 VIP 窗口的模拟结果

经过计算，得到下表。

表 2　增设VIP窗口前后模拟结果比较

到达队列情况	总用时（分钟）	1 号窗口繁忙程度	2 号窗口繁忙程度
P、P、V （增设 VIP 窗口前）	18	100%	50%
P、P、V （增设 VIP 窗口后）	18	50%	100%

3.1.2　第二问

运行bank_1_2 中的代码，设置银行营业时间为 50 分钟，VIP和普通窗口各 1 个，顾客到达事件间隔为 2 分钟，顾客服务时间为 8 分钟。得到在允许VIP顾客插队的情况下顾客的到达和离开情况，如下图所示。

```
Enter the simulation time in minutes: 72
Enter the number of bank tellers for VIP: 1
Enter the number of bank tellers for common: 1
Enter the range of arrival times in minutes: 2 2
Enter the range of service times in minutes: 8 8

********* Simulation Begin *********
Time:  0      arrival   customer    1     (P)    ⟹ Teller:    2
Time:  2      arrival   customer    2     (V)    ⟹ Teller:    1
Time:  4      arrival   customer    3     (P)    ⟹ Teller:    2
Time:  6      arrival   customer    4     (P)    ⟹ Teller:    2
Time:  8      departure customer    1     (P)    ⟸ Wait:      0     Service:      8
Time:  8      arrival   customer    5     (V)    ⟹ Teller:    1
Time: 10      departure customer    2     (V)    ⟸ Wait:      0     Service:      8
Time: 10      arrival   customer    6     (V)    ⟹ Teller:    2
Time: 12      arrival   customer    7     (V)    ⟹ Teller:    1
Time: 14      arrival   customer    8     (V)    ⟹ Teller:    2
Time: 16      departure customer    3     (P)    ⟸ Wait:      4     Service:      8
Time: 16      arrival   customer    9     (V)    ⟹ Teller:    1
Time: 18      departure customer    5     (V)    ⟸ Wait:      2     Service:      8
Time: 18      arrival   customer    10    (P)    ⟹ Teller:    2
Time: 18      departure customer    10    (P)      The customer gave up and left!
Time: 20      arrival   customer    11    (V)    ⟹ Teller:    2
Time: 22      arrival   customer    12    (P)    ⟹ Teller:    2
Time: 22      departure customer    12    (P)      The customer gave up and left!
Time: 24      departure customer    6     (V)    ⟸ Wait:      6     Service:      8
Time: 24      arrival   customer    13    (P)    ⟹ Teller:    2
Time: 24      departure customer    13    (P)      The customer gave up and left!
Time: 26      departure customer    7     (V)    ⟸ Wait:      6     Service:      8
Time: 26      arrival   customer    14    (V)    ⟹ Teller:    1
Time: 28      arrival   customer    15    (P)    ⟹ Teller:    2
Time: 28      departure customer    15    (P)      The customer gave up and left!
Time: 30      arrival   customer    16    (V)    ⟹ Teller:    2
Time: 32      departure customer    8     (V)    ⟸ Wait:      10    Service:      8
Time: 32      arrival   customer    17    (P)    ⟹ Teller:    2
Time: 32      departure customer    17    (P)      The customer gave up and left!
Time: 34      departure customer    9     (V)    ⟸ Wait:      10    Service:      8
Time: 34      arrival   customer    18    (P)    ⟹ Teller:    2
Time: 34      departure customer    18    (P)      The customer gave up and left!
Time: 36      arrival   customer    19    (P)    ⟹ Teller:    2
Time: 36      departure customer    19    (P)      The customer gave up and left!
Time: 38      arrival   customer    20    (V)    ⟹ Teller:    1
Time: 40      departure customer    11    (V)    ⟸ Wait:      12    Service:      8
Time: 40      arrival   customer    21    (P)    ⟹ Teller:    2
Time: 42      departure customer    14    (V)    ⟸ Wait:      8     Service:      8
Time: 42      arrival   customer    22    (P)    ⟹ Teller:    2
Time: 42      departure customer    22    (P)      The customer gave up and left!
Time: 44      arrival   customer    23    (P)    ⟹ Teller:    2
Time: 44      departure customer    23    (P)      The customer gave up and left!
Time: 46      arrival   customer    24    (P)    ⟹ Teller:    2
Time: 46      departure customer    24    (P)      The customer gave up and left!
Time: 48      departure customer    16    (V)    ⟸ Wait:      10    Service:      8
Time: 48      arrival   customer    25    (P)    ⟹ Teller:    2
Time: 50      departure customer    20    (V)    ⟸ Wait:      4     Service:      8
Time: 50      arrival   customer    26    (P)    ⟹ Teller:    2
Time: 50      departure customer    26    (P)      The customer gave up and left!
Time: 56      departure customer    4     (P)    ⟸ Wait:      42    Service:      8
Time: 64      departure customer    21    (P)    ⟸ Wait:      16    Service:      8
Time: 72      departure customer    25    (P)    ⟸ Wait:      16    Service:      8

******** Simulation Summary ********
Simulation of 72 minutes
No. of Common Customers:  5
No. of Vip Customers:  10
No. of Customers:  15
Average Common Customer Wait:   16 minutes
Average Vip Customer Wait:   7 minutes

********* Tellers Busyness *********
Teller 1 Working: 67%
Teller 2 Working: 100%
```


图 5　　允许 VIP 顾客插队情况下的模拟结果

由上图可以看出，第 6 分钟到达的顾客 4（P），先后被VIP顾客 6、8、11、16 插队，使其等待时间达到了 42 分钟。顾客 10、12 等多名顾客在到达银行后发现其等待时间过长，选择放弃排队，直接离开。

最终普通顾客的平均等待时间为 16 分钟，远高于VIP顾客的 7 分钟；1 号窗口的繁忙程度为 67%，远低于 2 号窗口的 100%，其原因是许多VIP顾客在普通窗口插队所需的等待时间比在VIP窗口更低，因而选择在 2 号窗口插队，导致了 1 号窗口的空置。

3.2 第 2 题：VIP 限时等待

运行bank_2_3 中的代码，设置银行营业时间为 50 分钟，VIP窗口 1 个，普通窗口 2 个，顾客到达事件间隔为 2-3 分钟，顾客服务时间为 5-15 分钟，VIP顾客最大容忍时间为 5 分钟。得到在VIP限时等待的情况下顾客的到达和离开情况，如下图所示。

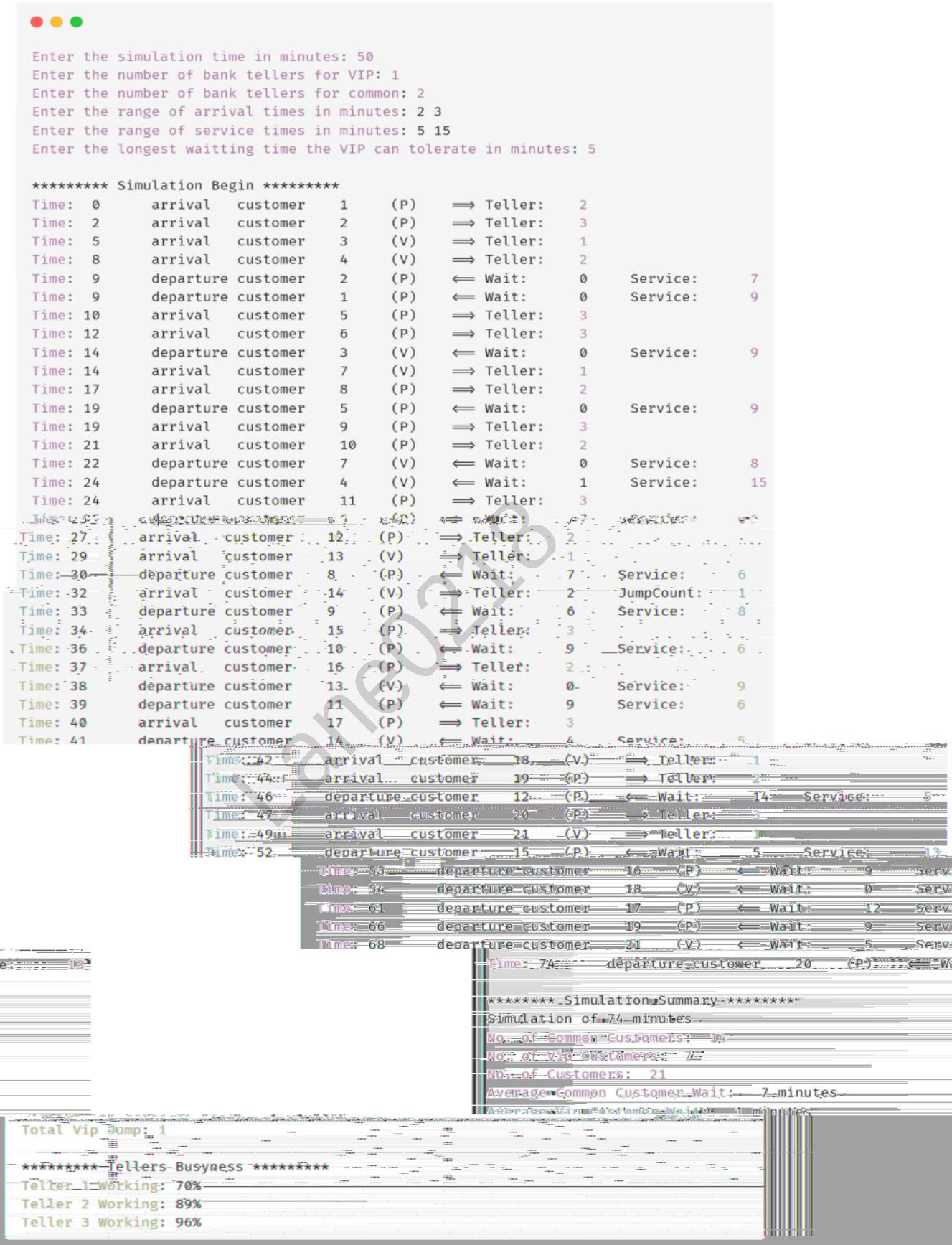

图 6　VIP 限时等待情况下的模拟结果

由上图的结果可以看出，VIP顾客 14 插了普通顾客 12 的队，从而使其等待时间降低到了 4 分钟，在最大容忍时间之内。对比上一问的结果发现，在当前排队机制下，VIP顾客并不能一到达就插队，而是要在其不插队的情况下等待时间将超过最大容忍时间时才允许插队。因此，VIP顾客的插队情况较少，普通顾客的等待时间不会太长，其不满意度下降；与此同时，更多VIP顾客选择在VIP窗口排队，因此VIP窗口的空置率下降，银行的不满意度下降。

3.3 第 3 题：最佳排队机制

3.3.1 最大容忍时间与总体不满意度之间的关系

运行bank_2_3 中的代码，设置银行营业时间为 50 分钟，VIP窗口 1 个，普通窗口 2 个，顾客到达事件间隔为 1-3 分钟，顾客服务时间为 5-20 分钟，VIP顾客最大容忍时间为 10 分钟。得到在VIP限时等待的情况下银行、普通顾客、VIP顾客以及总体的不满意度。

```
Enter the simulation time in minutes: 50
Enter the number of bank tellers for VIP: 1
Enter the number of bank tellers for common: 2
Enter the range of arrival times in minutes: 1 3
Enter the range of service times in minutes: 5 20
Enter the longest waitting time the VIP can tolerate in minutes: 10

******** Simulation Summary ********
Simulation of 124 minutes
No. of Common Customers:  16
No. of Vip Customers:  8
No. of Customers:  24
Average Common Customer Wait: 33 minutes
Average Vip Customer Wait: 5 minutes
Total Vip Jump: 9

********* Tellers Busyness *********
Teller 1 Working: 44%
Teller 2 Working: 97%
Teller 3 Working: 98%

****** Overall Dissatisfaction *******
Bank Dissatisfaction: 0.20
Common Customer Dissatisfaction: 0.32
VIP Customer Dissatisfaction: 0.00
Overall Dissatisfaction: 0.17
```

图 7 程序输入的参数和输出结果

在其余变量保持不变的情况下，修改VIP顾客最大容忍时间，在同一输入下多次运行程序取平均值，得到最大容忍时间与总体不满意度间的关系如下表所示。

表 3　最大容忍时间与总体不满意度之间的关系

最大容忍时间（分钟）	第一次	第二次	第三次	平均值
4	0.63	0.79	0.66	0.69
6	0.44	1.37	0.50	0.77
8	1.38	1.15	0.29	0.94
10	0.17	0.28	0.27	0.24
12	1.08	0.24	0.19	0.50
14	0.13	0.12	0.20	0.15
16	0.13	0.11	0.15	0.13

将上表数据绘制成散点图，并拟合出函数图像，如下图所示。

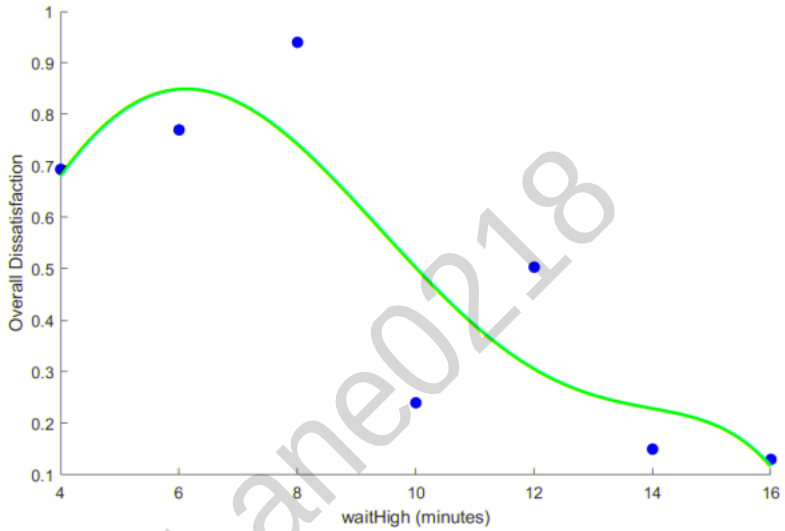

图 8　最大容忍时间与总体不满意度之间的关系图

由上述数据可以看出，总体不满意度大体上随最大容忍时间增加而先增大后减小。其原因是在最大容忍时间较小时，VIP顾客即使插队也不能将等待时间降低到最大容忍时间以内，所以选择不插队，因而银行和普通顾客的不满意度较低，总体不满意度较低。而在最大容忍时间较大时，VIP顾客即使不插队，等待时间也不会超过最大容忍时间，所以同样选择不插队；与此同时，普通顾客能忍受的等待时间也较大，因此总体不满意度较低。

通过多次运行程序发现，即使输入同样的参数，每次输出的总体不满意度也会有较大的差异，其原因是VIP顾客的顺序和数量是随机生成的，这将极大地影响最终的结果。因此即使固定输入的参数，本题也不可能得到一个“万金油”式的最佳排队机制。

3.3.2 银行营业时间与总体不满意度之间的关系

运行bank_2_3 中的代码，设置银行营业时间为 50 分钟，VIP窗口 1 个，普通窗口 2 个，顾客到达事件间隔为 1-3 分钟，顾客服务时间为 5-20 分钟，VIP顾客最大容忍时间为 8 分钟。得到在VIP限时等待的情况下银行、普通顾客、VIP顾客以及总体的不满意度。

在其余变量保持不变的情况下，修改银行营业时间，在同一输入下多次运行程序取平均值，得到银行营业时间与总体不满意度之间的关系如下表所示。

表 4 银行营业时间与总体不满意度之间的关系

银行营业时间（分钟）	第一次	第二次	第三次	平均值
25	0.06	0.10	0.13	0.10
50	0.44	0.23	1.33	0.67
75	0.45	2.51	1.41	1.46
100	3.76	2.64	1.35	2.58
125	1.66	2.50	0.77	1.64
150	2.92	1.56	2.65	2.38

同样，将其绘制成散点图，并拟合出函数图像，如下图所示。

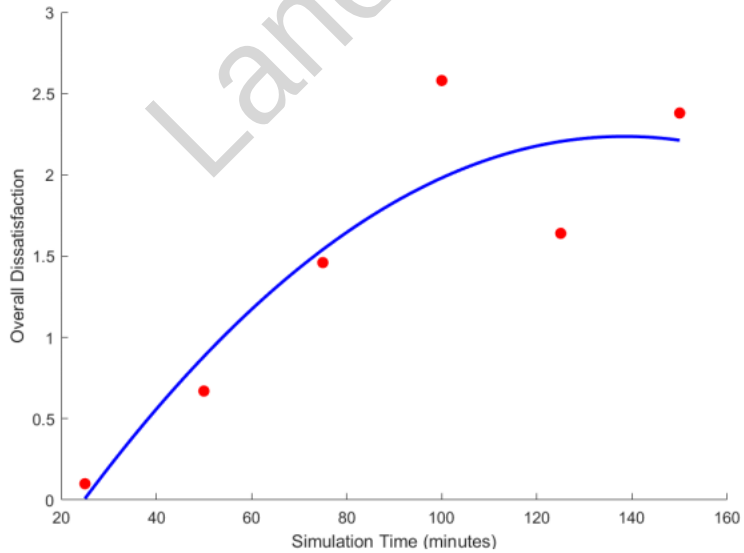

图 9 银行营业时间与总体不满意度之间的关系

由上述数据可以看出，随着银行营业时间的增加，总体不满意度呈现增大的趋势。

4 源代码解析

4.1 bank_1_1

本小题在框架代码的基础上修改了sim.h的部分代码，实现了对增设 1 个VIP窗口时顾客的到达和离开情况的模拟。

1) Event

增加属性isVip，用于表示顾客类型，其值为 1 表示顾客是VIP顾客，否则为普通顾客。

在构造函数中添加参数iv，用于初始化顾客类型。

添加int GetIsVip(void) const 函数，用于获取顾客类型数据。

2) Simulation

添加int NextAvailableTeller_vip(void) 函数，用于选择VIP顾客的排队窗口。由于本题设定 1 号为VIP窗口，2 号为普通窗口，故其选择逻辑是：当 1 号窗口的finishService 不为 0，且 2 号窗口的finishService为 0 时，选择 2 号窗口；其余情况选择 1 号窗口。

修改int NextAvailableTeller(void) 函数，用于选择普通顾客的排队窗口。本小题中普通顾客只能选择 2 号窗口。

修改void RunSimulation(void) 函数，用于从优先级队列中取出一个事件，进行处理并输出相应结果。在其中添加变量nextVip，用于指定下一个顾客的类型。此处根据“顾客类型顺序为P、P、V”的约定，用三元表达式计算nextVip：当下一个顾客为 3 号顾客时，设置为VIP顾客；否则设置为普通顾客。在选择窗口时，针对VIP和普通顾客要分别调用上述两个函数进行选择。

4.2 bank_1_2

4.2.1 apqueue.h

添加int judge(DataType e, int tellerID, int arrive)函数，用于判断该顾客能否被VIP顾客插队，可以被插队，返回 1；否则，返回 0。其判断逻辑是：对于其普通窗口的普通顾客，如果比VIP顾客先到达银行，并且在VIP

顾客到达时还没有开始服务，那么该普通顾客可以被VIP顾客插队。对于该普通顾客的到达时间，可以用其离开时间-服务时间-等待时间计算；对于其开始服务时间，可以用离开时间-服务时间来计算。

添加`int VIP_WaitTime_Reduce(int tellerID, int arrive)` 函数，用于计算通过在该普通窗口插队，VIP顾客的等待时间可以减少多少。对于该窗口的所有顾客，使用上述judge函数判断该顾客能被插队后，统计其服务时间，最终得到所有被插队的普通顾客服务时间总和。通过插队，该VIP顾客的等待时间就可以减少这个数值。在后续模拟的过程中，通过遍历各普通窗口，得到在每个窗口插队能减少的等待时间，可以选出等待时间最短的窗口。

添加`void Change_Common_Departure(int tellerID, int arrive, int wait)` 函数，在确定该VIP顾客在哪个普通窗口插队后，用于修改被插队的普通顾客离开事件的信息。用上述judge函数判断该普通顾客能被插队后，将其等待时间和离开时间均加上VIP顾客的服务时间。

添加`int Jump_Cnt(int tellerID, int arrive)` 函数，用于计算VIP顾客插了多少个普通顾客的队。同样用上述judge函数判断该普通顾客是否能被插队，统计插队数量。在后续模拟的过程中，修改窗口结构体的信息时，如果VIP顾客插队，那么其总等待时间要加上VIP顾客的等待时间，再加上被插队的普通顾客数量乘以VIP顾客的服务时间。

4.2.2 sim.h

1) 全局

定义全局变量TIMEG为3，表示当等待时间大于服务时间上限的3倍，顾客放弃排队，直接离开。

增加变量vipCustomers和vipWait，并初始化为0，用于统计VIP顾客的数量和VIP顾客总等待时间。

增加变量VIP_RATIO，用于表示VIP顾客所占比例，默认为30%。便于后续使用三元表达式`nextVip = (rand() % 100 - 1) > VIP_RATIO ? 0 : 1` 随机生成下一个顾客的类型。

2) Event

增加属性vip和函数int GetVip(void) const，与上文功能相同，不再赘述。

添加void SetTime(int vip_service_time)和void SetWaitTime(int vip_service_time)函数，用于修改普通顾客到达信息，这两个函数会被上述Change_Common_Departure函数调用。

3) Simulation

修改属性numTellers为numTellers_vip和numTellers_common，分别表示VIP和普通窗口的数量。

增加属性lose，用于统计有多少顾客放弃排队，直接离开.

修改int NextAvailableTeller(void) 函数，用于选择普通顾客的排队窗口。具体实现上，函数遍历各普通窗口，选择其中finishService最短的窗口排队。

添加int NextAvailableTeller_vip(void) 函数，用于选择VIP顾客的排队窗口。其选择逻辑是本小题的重点：遍历所有窗口，分别计算在VIP窗口排队和在普通窗口插队所需的等待时间，选取等待时间最短的窗口。

修改void RunSimulation(void) 函数，用于从优先级队列中取出一个事件，进行处理并输出相应结果。在其中添加变量nextVip，用于指定下一个顾客的类型。在选择窗口时，针对VIP和普通顾客要分别调用上述两个函数进行选择。如果VIP顾客选择去普通窗口插队，则先将该普通窗口的finishService临时减去VIP顾客插队降低的排队时间，从而能统一算出该VIP顾客的等待时间，随后再将其加回来；还需要调用函数更新被该VIP顾客插队的普通顾客的离开信息；并且，在更新窗口总等待时间信息时，不仅要加上该VIP顾客的等待时间，还要加上被插队的普通顾客数量乘以VIP顾客的服务时间。

如果顾客到达后发现等待时间大于服务时间上限的TIMES倍，放弃排队，直接离开。具体实现上，先将上述更新后的窗口信息、全局统计信息等减去当初加的数值，回到该顾客到达前的状态，然后添加该顾客的离开事件，设置其离开时间等于到达时间，设置其等待时间、服务时间均为0。否则，添加顾客的离开事

件，设置离开事件时间为该窗口finishService的时间，等待时间和服务时间均为上面计算出的数值。

RunSimulation和PrintSimulationResults函数中的输出语句均做了少量改动，用于输出增加的属性和变量情况，使程序的输出更清晰明了，此处不再赘述。

4.3 bank_2_3

4.3.1 apqueue.h

修改void PQInsert(const DataType &item)函数，在插入事件后调用sort语句进行排序，使pqlist队列按照时间顺序从先到后排列。

添加int Max_Jump_Index(int tellerID)函数，用于返回最多能插队到pqlist[3]的VIP之后，已在办理业务的顾客之后（取二者靠后的）。在VIP顾客插队时，只考虑pqlist中的离开事件，由于pqlist已经按时间顺序进行了排序，所以当pqlist[3]所对应的顾客不能被插队，那么下标为0-2对应的顾客也必定不能被插队。具体实现上，首先对pqlist从后往前找最后一个VIP顾客在pqlist的下标；如果没有找到VIP顾客，那么从前往后找正在服务的普通顾客，返回其下标。

添加int Jump_count(int waitHigh, int tellerID, int cur_finishService, int cur_time)函数，用于返回在该窗口要使得VIP等待时间不超过最大容忍时间需要插队的次数；如果返回值为-1，表示无法使等待时间降低到最大容忍时间。具体实现上，对pqlist从后往前遍历，直到上面函数求出的“最多能插队到的位置对应pqlist的下标”，依次将等待时间减去被插队的普通顾客服务时间，当等待时间降低到最大容忍时间以内，跳出循环。如果循环结束，等待时间仍然高于最大容忍时间，则返回-1，表示不插队。

oid Jump(int waitHigh, int tellerID, int vip_time, int nt, int &cur_finishService)函数，在确定VIP顾客在哪个窗口插后，更新被插队顾客的离开信息，并返回该VIP顾客的等待时间。

添加v
minJumpC
队多少次后

im.h

nk_1_2 中已有的或类似的变量和函数，在此处不再赘述。

4.3.2 s

在bar

1) 全局

增加变量totalJumpCnt并初始化为0，用于统计VIP顾客总共插了多少个普通顾客的队，在第3题中可以作为普通顾客不满意度的考虑因素。

添加bool cmp_greater_time(Event e1, Event e2)函数，用于对优先级队列排序。理论上完成排序后可以直接从前往后出队，不需要框架代码中重载运算符“<”的函数；此处为简便起见，仍然保留并修改框架代码中的重载运算符函数，其逻辑与本函数相同。

2) Event

添加属性jump，针对VIP顾客，表示是否插队，主要用于优先级队列的排序。

3) Simulation

添加属性waitHigh，针对VIP顾客，表示其最大容忍时间。

修改int NextAvailableTeller_vip(int arrive_time, int &isJump, int &minJumpCnt)函数，用于VIP顾客选择窗口。此函数的实现是解决本问题的关键。首先遍历各窗口（含VIP窗口），计算出在不插队的情况下，最少需要多少等待时间；若该等待时间不超过最大容忍时间，则返回该窗口，不插队；否则定义一个列表，用于储存在每个普通窗口插队最少需要插多少个人，才能使等待时间低于最大容忍时间，然后遍历各普通窗口，将数据储存到列表中；最后遍历该列表，找到插队次数最少的窗口；若列表中所有窗口的值均为-1，表示在任一普通窗口插队均不能使等待时间减少到最大容忍时间范围内，则不插队，仍然返回之前求出的等待时间最短的窗口。

修改void PrintSimulationResults(void)函数，用于输出新增的信息，并根据上文设计的不满意度公式计算并输出银行、普通顾客、VIP顾客不满意度以及总体不满意度，便于分析不同变量对总体不满意度的影响。