Project 3

家谱管理系统

班级：教务三班

彭高15331253

刘俊良 15331215

刘钟涛 15331222

## 「题目要求」

## 「数据结构与算法」

我们采用树状结构来模拟这个家谱模拟系统。我们定义了Person类来表示家谱中每一个人，定义Family类来表示家谱中不同家庭之间的关系。Person表示Family中的一个节点，Family类是一个二叉树。

最后，我们定义一个FamilyAgenda类来抽象整个管理系统，所有的操作都要通过FamilyAgenda来实现。

图1 Person类、Family类URL

#### Person类

1. 成员selfName表示Person的名字
2. 成员itself表示Person的性别
3. 指针partner表示他的配偶（可看作是二叉树中的左指针）
4. 指针child表示第一个儿子，或者是兄弟（可看作是二叉树的右指针）

#### Family类

1. 成员founder表示家族的建立者（可看作是二叉树的根节点）
2. 成员 familyName表示家族名
3. 成员members\_total表示家族总人数
4. 方法marry\_other用于模拟家族男性成员结婚
5. 方法born\_baby用于模拟家族生孩子
6. 方法remove\_person用于模拟家族中女性成员出嫁后被移出族谱
7. 方法find\_person为寻找名字为name的家族成员

#### FamilyAgenda类

1. 成员familyMap 为一个存放在当前系统中家族的键值队

2. 方法queryFamilyByName模拟寻找某个家族中名为name的人

3. 方法creatFamily 模拟家族创建

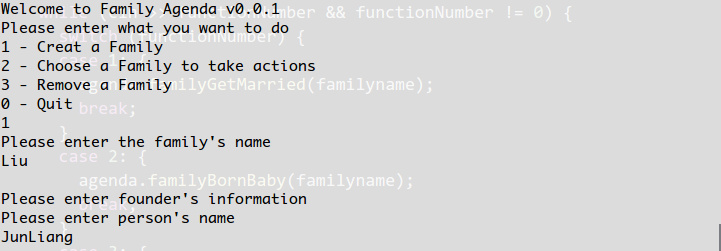
4. 方法removeFamily 模拟家族移除（本管理系统）

5. 方法familyBornBaby 模拟家族生孩子

## 「数据测试、结果及分析」

#### 家族创建

**[Input]**

图2 家族创建输入

**[output]**

图3 Problem1样例输出

表1 Problem1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **queue\_limit** | 5 | 6 | 8 | 10 | 15 | 18 |
| **expeted\_num** | .48/.48 | .48/.48 | .48/.48 | .48/.48 | .48/.48 | .48/.48 |
| **land\_refuese** | 0 | 0 | 0 | 0 | 0 | 0 |
| **takeoff\_refuse** | 38 | 29 | 19 | 13 | 3 | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **queue\_limit** | 5 | 5 | 5 | 5 | 5 | 5 |
| **expeted\_num** | .48/.48 | .45/.45 | .43/.43 | .4/.4 | .37/.37 | .33/.33 |
| **land\_refuese** | 0 | 0 | 1 | 1 | 1 | 0 |
| **takeoff\_refuse** | 38 | 31 | 13 | 10 | 3 | 0 |

1. 随着queue\_limit的增加，被拒绝的可能性降低
2. 随着expected number的降低，被拒绝的可能性降低

#### Problem2

**[Input]**

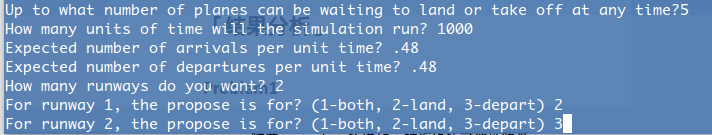


图4 Problem2样例输入

**[Output]**

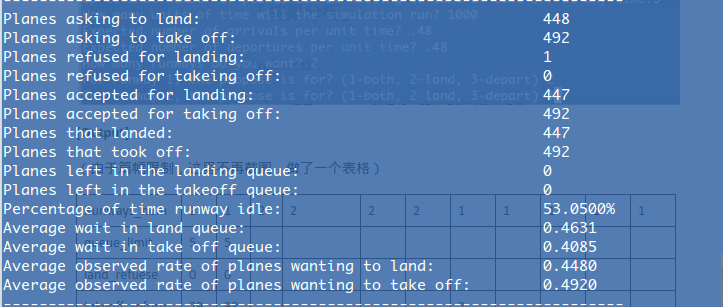


图5 Problem2样例输出

表2 Problem2

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Runways\_num** | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 |
| **queue\_limit** | 5 | 5 | 4 | 4 | 3 | 3 | 2 | 2 | 1 | 1 |
| **land\_refuese** | 1 | 0 | 5 | 5 | 10 | 10 | 27 | 27 | 85 | 85 |
| **takeoff\_refuse** | 0 | 29 | 0 | 28 | 4 | 73 | 26 | 105 | 96 | 171 |
| **landing** | 447 | 448 | 443 | 443 | 437 | 437 | 420 | 420 | 362 | 362 |
| **takeoff** | 492 | 463 | 492 | 462 | 488 | 419 | 466 | 387 | 396 | 321 |

1. 当queue\_limit比较大的时候，一条跑道和两条跑道能服务的飞机数量差别并不大。因为跑道可以容纳足够多的飞机而不需要两条跑道。
2. 但是当queue\_limit下降的时候（假设需求量不变），由于一条跑道容纳量有限，两条跑道优势会比较明显。

#### Problem3

**[Input]**

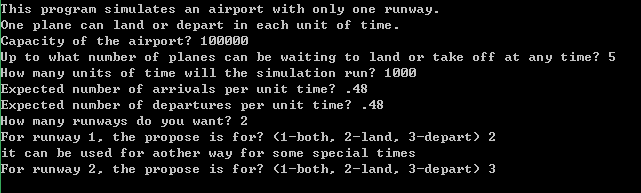


图6 Problem3样例输入

**[Output]**

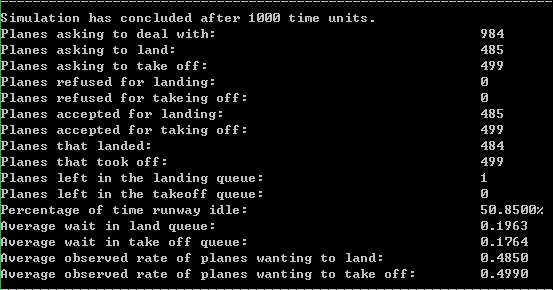


图7 Problem3样例输出

表3 Problem3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Runways\_num** | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| **queue\_limit** | 5 | 4 | 3 | 5 | 4 | 3 | 2 |
| **expected\_num** | .48/.48 | .48/.48 | .48/.48 | .35/.35 | .35/.35 | .35/.35 | .35/.35 |
| **land\_refuese** | 0 | 0 | 1 | 0 | 0 | 0 | 2 |
| **takeoff\_refuse** | 0 | 0 | 0 | 0 | 0 | 0 | 4 |
| **landing** | 485 | 340 | 484 | 340 | 340 | 340 | 338 |
| **Takeoff** | 499 | 365 | 499 | 365 | 365 | 365 | 361 |

对比problem2数据可得跑道的使用率增加

#### Problem4

**[Input]**

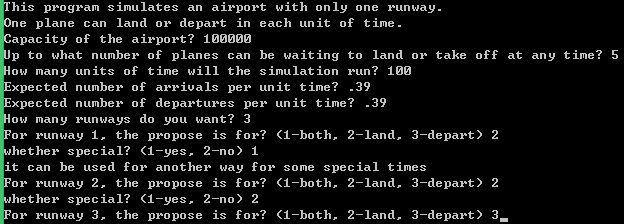


图8 Problem4样例输入

**[Output]**

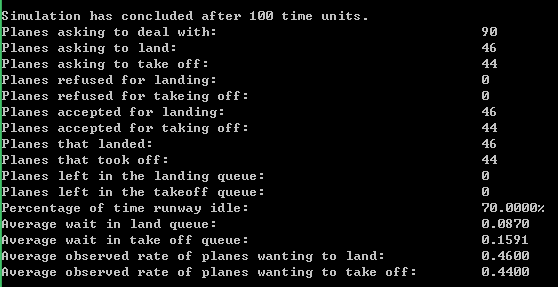


图9 Problem4样例输出

表4 Problem4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Runways\_num** | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| **queue\_limit** | 5 | 4 | 3 | 5 | 4 | 3 | 2 |
| **expected\_num** | .39/.39 | .39/.39 | .39/.39 | .25/.25 | .25/.25 | .25/.25 | .25/.25 |
| **land\_refuese** | 0 | 1 | 2 | 0 | 0 | 0 | 0 |
| **takeoff\_refuse** | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| **landing** | 46 | 45 | 44 | 31 | 31 | 31 | 31 |
| **Takeoff** | 44 | 44 | 44 | 36 | 36 | 36 | 34 |

#### Problem5

**[Input]**

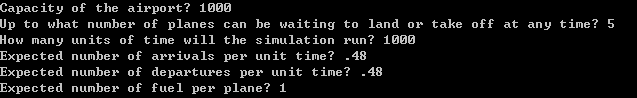


图10 Proble5样例输入

**[Output]**

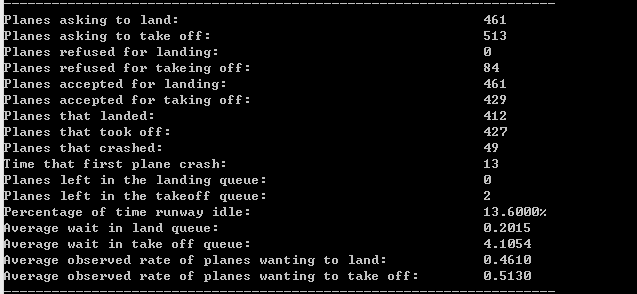


图10 Proble5样例输出

表5 Problem5

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **queue\_limit** | 2 | 3 | 4 | 5 | 5 | 5 |
| **expected\_num** | .48/.48 | .48/.48 | .48/.48 | .48/.48 | .4/.4 | .3/.3 |
| **expected\_fuel** | 1 | 1 | 1 | 1 | 1 | 1 |
| **plane\_crash** | 28 | 45 | 49 | 49 | 35 | 22 |
| **time\_first\_crash** | 63 | 13 | 13 | 13 | 16 | 31 |
| **number of planes in the queue when first plane crash** | 2 | 5 | 6 | 7 | 6 | 1 |

表6

|  |  |  |  |
| --- | --- | --- | --- |
| **queue\_limit** | 5 | 5 | 5 |
| **expected\_num** | .48/.48 | .48/.48 | .48/.48 |
| **expected\_fuel** | 2 | 3 | 5 |
| **plane\_crash** | 23 | 11 | 1 |
| **time\_first\_crash** | 14 | 4 | 416 |
| **number of planes in the queue when first plane crash** | 8 | 8 | 7 |

从测试数据可以看出，当queue limit比较小时，它的变大会使飞机坠落的开始时间和坠落总数变大，当queue limit达到4或者以上时，这个两个数据就不再变大了，这时应该是油量的限制，导致等待飞机过多但是却无法及时降落导致坠毁。而机场忙碌程度会随queue limit增大而增大。

在到达飞机减少的情况下，飞机坠毁的开始时间会变晚，坠落的总数也会变少，机场忙碌程度变低，这时落地的队列比较小，因此飞机等待时间较短，坠落的可能性变小。

在飞机平均油量增加的情况下，坠落的总数会变小，因为飞机有足够的油量等待降落，但是机场忙碌程度基本不变。

#### Problem6

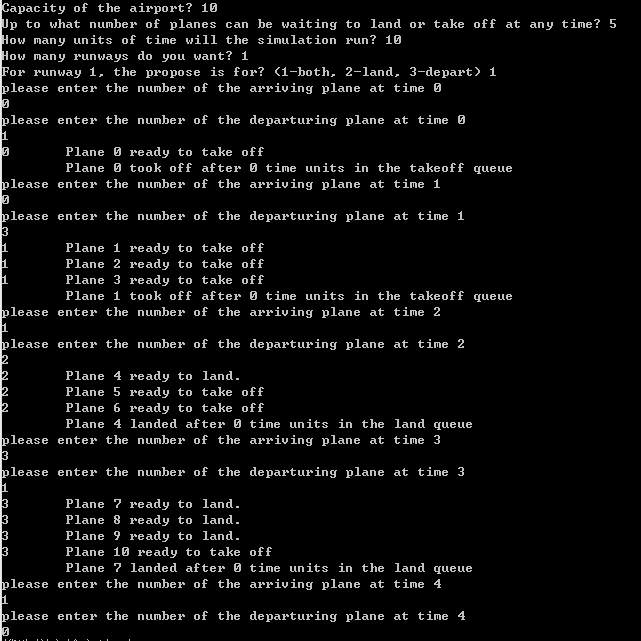


图11 Proble6样例输出

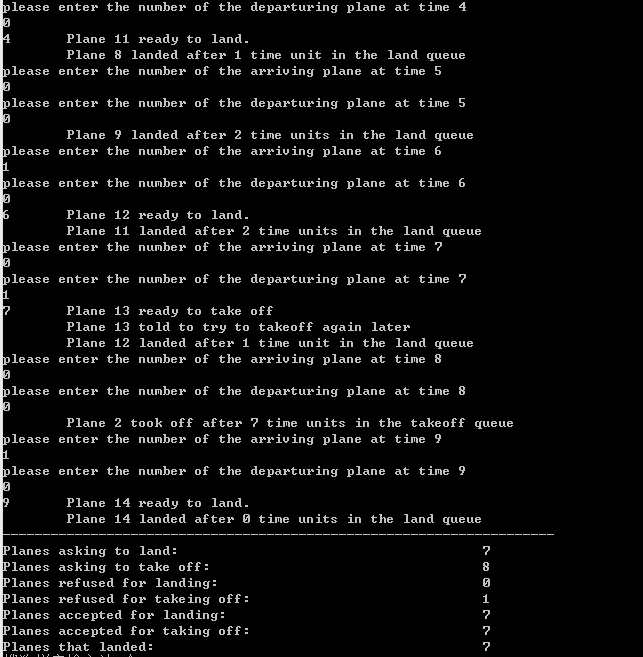


图12 Proble6样例输出



图13 Proble6样例输出

## 「分工、贡献%、自我评分」

彭高(33%)：负责完成可执行程序框架和类的改写。实验报告撰写。

刘俊良(33%)：完成书上题目。实验报告撰写。

刘钟涛(33%)：完成书上题目。程序可读性和输出细节调整。实验报告撰写。

## 「项目总结」

该项目模拟了飞机在跑道上起飞与降落的情景，利用起飞与降落两个队列完成调度，遵循现实中机场的实际运营来调用机场的跑道。在执行项目的过程中，因为要实现多跑道的调用，所以编写了一个机场类来包含其它，使结果更为合理。通过该项目我们可以进一步地研究对象的特性，分析其结构与操作，模拟出现实中的系统行为，从而便利其实用性和可靠性。

## 程序清单

Project2/

├── Problem1/

│ ├── Random.h

│ ├── main.cpp

│ └── Simulation.h

└── Problem2/

│ ├── Airport.h

│ ├── enum.h

│ ├── init.h

│ ├── Plane.h

│ ├── Runway.h

│ └── main.cpp

└── Problem3/

│ ├── Airport.h

│ ├── enum.h

│ ├── init.h

│ ├── Plane.h

│ ├── Runway.h

│ └── main.cpp

└── Problem4/

│ ├── Airport.h

│ ├── enum.h

│ ├── init.h

│ ├── Plane.h

│ ├── Runway.h

│ └── main.cpp

└── Problem5/

│ ├── Airport.h

│ ├── enum.h

│ ├── init.h

│ ├── Plane.h

│ ├── Runway.h

│ └── main.cpp

└── Problem6/

├── Airport.h

├── enum.h

├── init.h

├── Plane.h

├── Runway.h

└── main.cpp