

# 眼科病床的合理安排

## 摘 要

本文对眼科手术病床的安排与优化问题，建立了层次分析模型，基于 HRRN 算法的病床安排模型、灰色预测模型以及基于排队论的服务强度评价模型，解决了对医院目前病床安排模型的评价，病床的合理安排，病人入住时间的预测、手术安排时间的调整以及病床按比例分配的问题。

针对问题一，建立了层次分析模型，解决了对医院目前病床安排方式的评价问题。首先我们选择病床使用率、病床周转次数、等待住院时间满意度、等待手术时间满意度四个指标；其次建立层次分析模型，使用层次分析法中的判断矩阵计算各指标的权重得到综合评价值；最后对判断进行一致性检验。用此方法评价该医院现病床安排模型，得到的评价值为 0.6927。

针对问题二，建立了基于 HRRN 算法的病床安排模型，解决了病床的合理安排问题。我们先引入计算机操作系统作业调度中的 HRRN 算法，再综合考虑不同时间不同眼科疾病的优先级，并得出动态优先级函数和算法步骤，最后按动态优先级对病床进行安排。根据问题一建立的评价体系，该模型的评价值为 0.7967，高于医院现用的病床安排模型的评价值。

针对问题三，建立了灰色预测模型，解决了病人入住时间的预测问题。我们先建立灰色预测模型对未来 14 天每种眼科疾病到来的人数分别进行预测，再利用问题二的模型和算法计算出病人的入院时间，最后选择合理的分布函数求出置信区间。我们得到单眼白内障、双眼白内障、外伤、青光眼和视网膜类疾病患者的平均大致入住区间分别是 5-10 天、7-13 天、1-2 天、9-15 天、8-14 天。

针对问题四，在周六周日不安排手术的前提下，我们根据不同情况重新计算优先级并沿用问题二模型对问题进行求解。首先我们对问题进行分析得到三种备选方案，其次针对每一种备选方案我们重新计算每天每种疾病的优先级，最后利用问题二的模型再次求解并用问题一的模型进行评价。最终结果为白内障手术应该安排在周三和周五，该方案的综合评价值为 0.8344。

针对问题五，建立了基于排队论的服务强度评价模型模型，解决了病床按比例分配的问题。首先我们根据计算各类疾病的平均到达率和平均服务率；其次定义了系统服务强度，当各类病人的系统服务强度相同时，总的系统服务率达到最高；最后得到分配给单眼白内障、双眼白内障、外伤、青光眼和视网膜类疾病患者的病床数为 9、18、7、11、34。

**关键词：**层次分析模型；HRRN 算法；灰色预测模型；服务强度评价模型；

## 一、 问题重述

医院就医排队是大家都非常熟悉的现象，它以这样或那样的形式出现在我们面前，例如，患者到门诊就诊、到收费处划价、到药房取药、到注射室打针、等待住院等，往往需要排队等待接受某种服务。

我们考虑某医院眼科病床的合理安排的数学建模问题。

该医院眼科门诊每天开放，住院部共有病床 79 张。该医院眼科手术主要分四大类：白内障、视网膜疾病、青光眼和外伤。附录中给出了 2008 年 7 月 13 日至 2008 年 9 月 11 日这段时间里各类病人的情况。

白内障手术较简单，而且没有急症。目前该院是每周一、三做白内障手术，此类病人的术前准备时间只需 1、2 天。做两只眼的病人比做一只眼的要多一些，大约占到 60%。如果要做双眼是周一先做一只，周三再做另一只。

外伤疾病通常属于急症，病床有空时立即安排住院，住院后第二天便会安排手术。

其他眼科疾病比较复杂，有各种不同情况，但大致住院以后 2-3 天内就可以接受手术，主要是术后的观察时间较长。这类疾病手术时间可根据需要安排，一般不安排在周一、周三。由于急症数量较少，建模时这些眼科疾病可不考虑急症。

该医院眼科手术条件比较充分，在考虑病床安排时可不考虑手术条件的限制，但考虑到手术医生的安排问题，通常情况下白内障手术与其他眼科手术（急症除外）不安排在同一天做。当前该住院部对全体非急症病人是按照 FCFS（First come, First serve）规则安排住院，但等待住院病人队列却越来越长，医院方面希望你们能通过数学建模来帮助解决该住院部的病床合理安排问题，以提高对医院资源的有效利用。

问题一：试分析确定合理的评价指标体系，用以评价该问题的病床安排模型的优劣。

问题二：试就该住院部当前的情况，建立合理的病床安排模型，以根据已知的第二天拟出院病人数来确定第二天应该安排哪些病人住院。并对你们的模型利用问题一中的指标体系作出评价。

问题三：作为病人，自然希望尽早知道自己大约何时能住院。能否根据当时住院病人及等待住院病人的统计情况，在病人门诊时即告知其大致入住时间区间。

问题四：若该住院部周六、周日不安排手术，请你们重新回答问题二，医院的手术时间安排是否应作出相应调整？

问题五：有人从便于管理的角度提出建议，在一般情形下，医院病床安排可采取使各类病人占用病床的比例大致固定的方案，试就此方案，建立使得所有病人在系统内的平均逗留时间（含等待入院及住院时间）最短的病床比例分配模型。

## 二、 问题分析

### 2.1 问题一的分析

在医院现有资源的条件下解决评价病床安排模型的优劣的问题,此类题通常都是通过建立几个具有代表意义的指标来解决的。根据已知条件,采用层次分析法为宜,要用该方法解决,需要有各个层次的具体条件。因此,应有只考虑影响患者的主要因素的假设,有患者对于各项满意度的数据,在此基础上,可以采用层次分析法确定构成指标体系的四个指标,计算出权重并做判断矩阵的一致性检验。通过分析,选择层次分析法是最合理的。

### 2.2 问题二的分析

经过问题一的分析,在医院现行的 FCFS (First come, First serve) 规则条件下解决了制度不合理的问题。根据 FCFS 的条件,采用计算机操作系统中的最高响应比优先 (HRRN) 调度策略为宜。该策略通常通过计算优先级求解。根据 HRR 适用的条件,应对引入这种方法的科学性进行论证,再结合眼科的手术安排情况做出分析,通过各类疾病的优先级,最后得出结果第二天应安排哪些患者住院,并利用第一问的评价体系对其做出评价。

### 2.3 问题三的分析

在已知当时住院病人及等待住院病人的统计情况条件下解决了估计病人大致入住时间区间问题,这样的问题通常都是由预测的方法解决的。根据各类型患者一段时间内每天到达的人数,采用灰色预测的方法为宜。要用此方法解决,需要有原始数列、生成数列和一次累加数列。为了满足该条件,应有各类型患者一定时间内到达人数的数据,在此基础上,可以建立灰色预测模型,采用计算残差和级比偏差进行检验,得出各类型患者的大致入住区间。

### 2.4 问题四的分析

在周六周日不安排手术的条件下各类型患者何时入院接受治疗的问题,此问题与第二问采用的基本思想类似,根据第二问已知的条件,分情况列出备选方案。通过采用对备选方案每天各类型疾病优先级重新计算的方法,得出我们需要的患者术前等待时间的条件。再利用第一问的综合评价体系,得出每种方案的患者的满意度和合理的手术方案。

### 2.5 问题五的分析

在便于医院管理的条件下解决了病床比例大致固定的问题。根据各类患者所占的百分比,采用排队论的方法为宜。采用这种方法,我们需要有平均到达率,平均服务率的相关数据。在此基础上,可以建立服务强度平衡模型,最后通过系统服务率的高低,判断各类型疾病的分配床数是否合理。

### 三、 基本假设

- 1、所有数据来源于题目已知，真实可靠。
- 2、在考虑病床安排时可不考虑手术条件的限制。
- 3、假设住院期间不随意调动病人的安置情况。
- 4、假设将所有外伤都视为急症是合理的。
- 5、假设在出现的等待时间内病人的情况不会变得异常严重。

### 四、符号说明

|             |            |
|-------------|------------|
| $\alpha$    | 实际等待住院的时间  |
| $\beta$     | 做手术需要的标准时间 |
| $\gamma$    | 等待手术的时间    |
| $q$         | 表示使用的病床数   |
| $Q$         | 总病床数       |
| $A$         | 出院的人数      |
| $a$         | 表示医院空床数    |
| $S_i(1)$    | 等待住院时间的满意度 |
| $S_i(2)$    | 等待手术的时间满意度 |
| $S$         | 综合评价       |
| $C.I$       | 一致性指标      |
| $C.R$       | 随机性一致性比值   |
| $n$         | 病床的月周转次数   |
| $R_p$       | 响应比        |
| $\lambda_i$ | 平均到达率      |
| $\mu_i$     | 平均服务率      |
| $m_i$       | 病床数        |
| $\rho_i$    | 系统服务强度     |

### 五、模型的建立与求解

#### 5.1 数据分析

经过对该医院各类型疾病数据的分析，我们发现术前住院时间过长是当前病床使用率效率不高的主要因素，同时，入院后患者等待手术的时间也在很大程度上影响了患者对该医院的满意度。

为了方便计算，我们将不同类型的疾病定义如表 1 所示：

表 1 疾病的类型

| $i$  | 1       | 2       | 3  | 4   | 5      |
|------|---------|---------|----|-----|--------|
| 疾病类型 | 白内障（单眼） | 白内障（双眼） | 外伤 | 青光眼 | 视网膜类疾病 |

其次，对已出院每类病人人数和各项医疗时间进行统计：

表 2 已出院每类病人人数

| 类别      | 总人数（人） | 百分比   |
|---------|--------|-------|
| 白内障（单眼） | 72     | 20.6% |
| 白内障（双眼） | 82     | 23.5% |
| 外伤      | 55     | 15.8% |
| 青光眼     | 39     | 11.2% |
| 视网膜类疾病  | 101    | 28.9% |
| 合计      | 349    | 100%  |

表 3 已出院病人各项医疗时间

| 类别      | 术前等待时间（天） | 术后观察时间（天） | 住院等待时间（天） | 住院时间（天） | 总医疗时间（天） |
|---------|-----------|-----------|-----------|---------|----------|
| 白内障（单眼） | 2.3       | 3.0       | 12.5      | 5.3     | 17.8     |
| 白内障（双眼） | 3.6       | 3.0       | 12.5      | 8.5     | 21.7     |
| 外伤      | 1.0       | 6.0       | 1.0       | 7.0     | 8.0      |
| 青光眼     | 2.4       | 8.1       | 12.3      | 10.5    | 22.7     |
| 视网膜类疾病  | 2.4       | 10.3      | 12.5      | 12.7    | 25.2     |

整理数据如图 1、图 2、图 3、图 4 所示：



图 1 单眼白内障

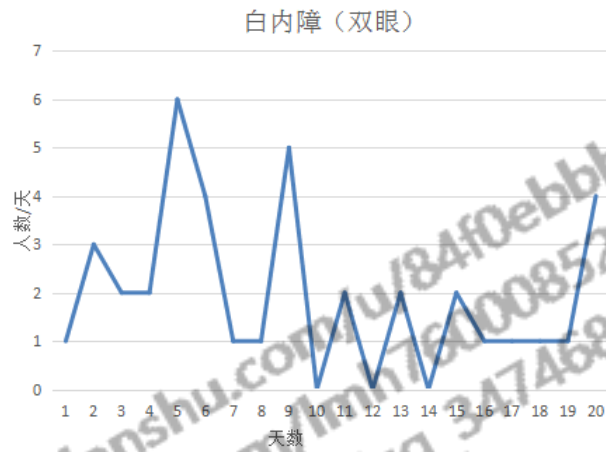


图 2 双眼白内障



图 3 外伤

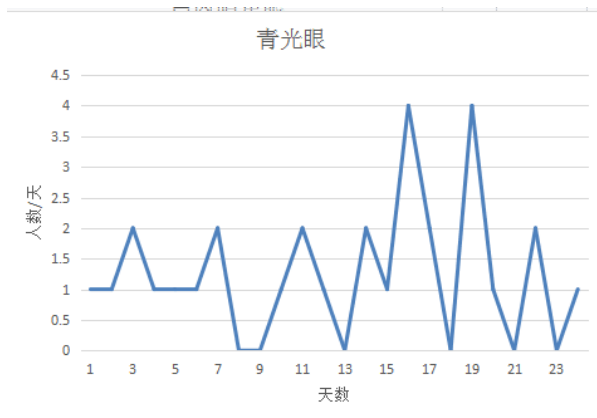


图 4 青光眼



图 5 视网膜类疾病

## 5.2 问题一的模型建立与求解

要确定眼科病床安排合理的评价指标体系,为了兼顾患者与医院双方的利益,需要从患者总体的满意度和病床安排的合理性两方面来考虑。患者等待住院的时间和等待手术的时间都可以影响患者的满意度;而病床的利用率基于病床使用率和病床的周转次数。我们决定用患者等待住院时间的满意度、等待手术的时间满意度、病床使用率和病床周转次数四个指标构成评价指标体系。通过层次分析法计算出各个指标的权重以得出综合评价值,最后用矩阵的一致性检验对评价指标体系的合理性做出判断。

### 5.2.1 模型的建立

想要建立合理的评价综合指标,必须要充分利用医院的资源,并且最大程度的满足患者的需要。因此,我们从两个方面考虑模型建立,一个是基于患者满意度的模型,另一个是基于病床利用率的模型。

患者的满意度取决于等待住院时间、等待手术时间和术后观察时间,等待住院时间和等待手术时间越长患者越不满意,因此我们抽取患者等待住院的时间和等待手术的时间作为衡量患者满意度的标准。充分利用医院的资源则考虑病床的利用率以及病床的周转次数。即 79 张床的空床尽可能少和在一段时间内出院人数尽可能多。

综上所述,我们纳入评价体系的指标共有 4 个,即患者等待住院时间的满意度、等待手术的时间满意度、病床使用率和病床周转次数。

#### (1) 基于患者满意度的模型建立

对于每位患者来说,在等待治疗的过程中,在不存在医院主观的延误现象时,影响其满意度的因素主要体现在实际等待住院的时间  $\alpha$ 、做手术需要的标准时间  $\beta$ , 以及等待手术的时间  $\gamma$ 。患者等待住院时间=实际住院时间-就诊时间,等待手术时间=第一次手术时间-住院时间。

##### ① 等待住院时间的满意度:

在理想的情况下,当患者门诊时确定需要住院治疗时,医院应尽快让患者住院,以保证最好的治疗效果,以免患者等待时间过长而造成不满。

由此建立等待住院时间的满意度  $S_1(1)$  的方程:

$$S_1(1) = \begin{cases} 1 & \alpha \leq \beta \\ \frac{\beta}{\alpha} & \alpha > \beta \end{cases}$$

② 等待手术时间满意度：

根据病情的不同，手术准备的时间也不同，但是由于手术时间的限制可能会使实际准备手术的时间超过需要准备手术的时间，同样影响患者的满意度。手术前等待时间越短，患者满意度越高。由此建立等待手术的时间满意度方程  $S_1(2)$ ：

$$S_1(2) = \begin{cases} 1 & \gamma \leq \beta \\ \frac{\beta}{\gamma} & \gamma > \beta \end{cases}$$

(2) 基于病床利用率的模型建立

病床的利用率主要是从医院的利益来考虑，即从病床使用率和病床周转次数两方面入手。较高的使用率和周转率不仅提高了医院的服务质量，还增加了收益。

① 病床的使用率：

为了最大程度的利用医院的资源，要尽可能的避免患者较少而空床位过多的现象，因而病床的使用率直接反应了病床的利用率。它是指当前使用的病床数占总病床数的百分比。我们用  $q$  表示使用的病床数， $Q$  表示总病床数，进而得到病床使用率  $S_2(1)$ ：

$$S_2(1) = \frac{q}{Q} \times 100\%$$

② 病床周转次数：

病床周转次数反映的是每张病床服务的患者数量，是一定时期内出院的人数与医院空床数之比。我们用  $A$  表示出院的人数， $a$  表示医院空床数，进而得到病床周转次数  $S_2(2)$ ：

$$S_2(2) = \frac{A}{a} \times 100\%$$

综合上述指标，患者的总体满意度  $S$  可以表示为：

$$S = \sum_{i=1}^2 \sum_{j=1}^2 S_i(j)$$

(3) 层次分析法模型的建立

① 建立层次结构模型

我们对病床的评价指标体系构造层次分析模型如图 1 所示。目标层  $U$  为眼科病床的合理安排，准则层分为  $B_1$ 、 $B_2$  两个准则，指标层我们根据准则层的共选取四个指标  $C_{11}$ 、 $C_{12}$ 、 $C_{21}$ 、 $C_{22}$  分别是基于患者满意度准则  $B_1$  下的等待住院时间的满意度、等待手术的时间满意度；基于病床利用率准则  $B_2$  下的病床使用率和病床周转次数。



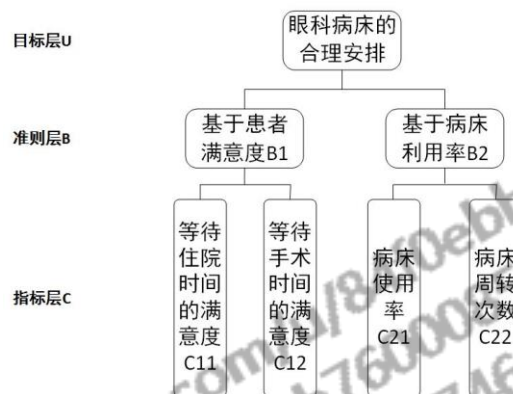


图 6 眼科病床的合理安排

## ② 构造判断矩阵

$A = (a_{ij})$  为判断矩阵, 用来表示同一层次各个指标相对重要性的判断值。

层次分析法中通过对指标之间两两重要程度进行比较分析和判断, 在对于两个不能准确量化的量作比较时, 常采用 1~9 的比较尺度:

同等重要为 1, 重要为 3, 很重要为 5, 非常重要为 7, 极重要为 9; 不重要为 1/3, 很不重要为 1/5, 非常不重要为 1/7, 极不重要为 1/9, 处在以上各等级之间时, 可记为 2, 4, 6, 8 或 1/2, 1/4, 1/6, 1/8。

## ③ 一致性检验

在得到了判断矩阵  $A$  后, 可以求得  $A$  的最大特征根  $\lambda_{\max}$ , 为了判断  $A$  的不一致程度是否在允许的范围内, 需要对它的一致性进行检验。

一致性指标:

$$C.I = \frac{\lambda_{\max} - n}{n - 1}$$

由于随着  $n$  的增加判断误差就会增加, 因此判断一致性时应当考虑到  $n$  的影响, 使用随机性一致性比值:

$$C.R = \frac{C.I}{R.I}$$

$R.I$  为平均随机一致性指标。当  $C.R=0$  时, 表示  $A$  通过一致性检验;  $C.R$  越大, 则说明判断矩阵的一致性越差; 但当  $C.R < 0.1$  时,  $A$  的不一致程度在允许范围内, 否则应重新考虑并调整两两比较判断矩阵, 直到具有满意的一致性。

表 4 随机一致性指标  $R.I$  的数值

| $n$   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| $R.I$ | .58 | .90 | .12 | .24 | .32 | .41 | .45 | .49 |

## 5.2.2 模型的求解

(1) 对患者各项指标的求解：

首先我们对每个类型病人统计其等待住院时间和等待手术时间的满意度如表 5 所示：

表 1 各类型病人的满意度

| 类别      | 等待住院时间满意度 | 等待手术时间满意度 |
|---------|-----------|-----------|
| 白内障（单眼） | 0.424     | 0.71      |
| 白内障（双眼） | 0.535     | 0.52      |
| 外伤      | 1         | 1         |
| 青光眼     | 0.844     | 1         |
| 视网膜类疾病  | 0.924     | 0.93      |

最终我们得出病床使用率为 0.996；

对题目中给出的数据进行分析，我们发现可以将 7 月 13 日往后的住院人数视为新增住院人数，病床周转次数与 7 月 12 日之前的住院人数无关。因此，我们将 8 月 8 日至 9 月 11 日作为计算病床周转次数的时期。经统计，这段时间内出院的人数有 302 人，以每个月 30 天计算病床的月周转次数，即：

$$n = \frac{302}{79} \times \frac{30}{34} = 3.373$$

$$n_{\max} = \frac{30}{5.3} = 5.66$$

即该医院病床的平均月周转次数为 3.373，最大月周转次数为 5.66。

标准化后的月病床周转次数：

$$m = \frac{n}{n_{\max}} = \frac{3.373}{5.66} = 0.596$$

(2) 判断矩阵和权向量的求解

设准则层包含 4 个准则，即 C1：患者等待住院时间的满意度；C2：等待手术的时间满意度；C3：病床使用率；C4：病床周转次数。通过查阅相关文件，得到判断矩阵 A 如下：

$$A = \begin{bmatrix} 1 & 3 & 3 & 1/1.5 \\ 1/3 & 1 & 1 & 1/4.5 \\ 1/3 & 1 & 1 & 1/4.5 \\ 3 & 4.5 & 4.5 & 1 \end{bmatrix}$$

我们得出权向量  $w_0 = (0.2840, 0.0947, 0.0947, 0.5266)$ ；

由此，综合评价价值  $S=0.6927$ 。说明医院目前的 FCFS 规则能满足一定的需求，但是由于等待的时间越长，总体满意度呈现下降的趋势，所以此方法有待改进。

## 5.2.3 模型的检验与分析

一致性检验得到的结果如表 2 所示：

表 2 一致性检验的结果

| 权向量 | 最 | C. I | C. R | 矩阵一致性 |
|-----|---|------|------|-------|
|-----|---|------|------|-------|

|  | 大<br>特<br>征<br>值 |            |            | 是否可以接受 |
|--|------------------|------------|------------|--------|
| $w_0 = (0.2840, 0.0947, 0.0947, 0.5266)$ | 4.2<br>361       | 0.0<br>787 | 0.0<br>884 | 是      |

因为  $C.R=0.0884 < 0.1$ ，因此 A 的不一致程度在允许范围内，矩阵的一致性是可以接受的。

因此从两个方面出发而建立的评价指标体系是相对合理的。它兼顾了患者和医院双方的利益；从患者的角度来说，患者自然希望能够尽早的入院接受治疗，反映在数据中即为患者的等待住院时间越短越好；从医院的角度来说，床位利用是否合理也是综合评判的重要指标，即病床的使用率和周转次数。

### 5.3 问题二的模型建立与求解

经过问题一的分析，我们发现了医院现行的 FCFS (First come, First serve) 规则的不合理性并引入计算机操作系统中的最高响应比优先 (HRRN) 调度策略。首先，我们对引入这种方法的科学性进行论证，其次结合眼科的手术安排情况做出分析，得出各类疾病的优先级，最后得出结果第二天应安排哪些患者住院，并利用第一问的评价体系对其做出评价。

#### 5.3.1 模型的建立

经过分析医院现行的 FCFS (First come, First serve) 规则，虽然是多数医院采用的方法，但它实际上存在一定的不合理性。FCFS 规则并没有考虑到患者病情的严重程度和病床的使用率，即它没有考虑到外伤的患者需要立即住院治疗，而且过于偏袒先来的病人，使得有的患者不能及时手术而在医院空等，使得病床的利用率下降。

为了得到更理想的病床安排模型，我们选用处理机中的最高响应比优先 (HRRN) 调度策略。FCFS 规则考虑的只是患者等待的时间，而忽视了患者住院的持续时间。而最高响应比优先 (HRRN) 调度策略就很好的解决了这个问题，既考虑了等待时间，又考虑了运行时间，使医院的病床安排更加合理。

优先级的计算公式为：

$$\text{优先级} = \frac{\text{等待时间} + \text{服务时间}}{\text{服务时间}}$$

由于等待时间与服务时间之和就是系统对该作业的响应时间，故该优先级又相当于响应比  $R_p$ 。据此，优先又可以表示为：

$$R_p = \frac{\text{等待时间} + \text{服务时间}}{\text{服务时间}} = \frac{\text{响应时间}}{\text{要求服务时间}}$$

由此可以得出：

(1) 如果作业的等待时间相同，则要求服务的时间越短，其优先权越高，因而类似于 SJF (Short job first) 算法，有利于短作业；

(2) 当要求服务的时间相同时，作业的优先权又决定于其等待时间，该算法类似于 FCFS 算法；

(3) 对于长作业的优先级，可以随等待时间的增加而提高，而其等待时间是

够长时，也可获得处理机。

因此该算法实现了较好的折中。我们针对医院眼科安排的具体情况，对 HRRN 进行了必要的改进。

接着，我们将对引入计算机操作系统的合理性进行论证：

在多道批处理系统中，有多个作业等待执行时，而一个处理器在每一时刻只能让一个作业占用，因此整个系统只能安排某些作业占用已经空闲的处理机。因此，系统必须依据某个策略做出选择，即决定在一个特定时间运行某几个作业，这个策略称为“处理机调度策略”。又因其有的目标相互冲突，有的目标相互补充，因此，判断处理器应该在什么时候分配给什么进程，必须进行综合性的考虑。

不难看出，处理器对作业的调度过程与医院病床的安排问题是极为相似的。医院的病床数相当于系统资源，而各种患者对应于等待处理机的作业，这样，在不考虑手术时间时，病床的合理安排问题可以转化为作业问题。下面给出两种问题的合理性论证：

(1) 服务对象的一致性：

每个作业的预期服务时间是不同的，不同的作业需要的资源也是不同的。对应到患者中，我们可以认为不同类型的患者的需要的资源不同，不同类型的疾病预期服务的时间也不同，也有优先级的区别。因此，等待处理机的进程与等待病床的患者具有一致性。

(2) 调度目标的一致性：

处理机调度策略和病床的安排都需要达到一定的目标，如达到最大吞吐量（即一定时间内出院人数多）、延时时间最短（即患者等待住院的时间短）、达到处理器利用率最大化（即病床利用率高）等等。可见，在处理器调度中，如果达到了它要求的目标，也就达到了合理病床安排的目标。

(3) 实现方法的一致性：

实际上就是指处理机调度中提出的解决方法可应用于病床安排问题，两者在实现方法上式可以相互转化的，即可用于作业调度的方法同样可适用于解决病床安排的问题，这保证了我们的工作效率。

综上所述，我们认为可以使用处理器调度的有关方法来解决病床合理安排的问题。在这一领域已经有了很多成熟的算法，因此转化的关键在于如何在处理器调度策略中考虑手术安排时间。

接下来我们把处理机调度算法的目标与医院眼科病床安排的具体情况相结合，从如下几个方面进行分析：

(1) 处理机调度算法的共同目标：

① 资源利用率：

提高系统的资源利用率就是使系统中的资源尽可能的保持忙碌状态，即提高病床的利用率。

② 公平性：

指应使各个进程都获得合理的时间，对相同类型的进程应获得相同的服务；对不同类型的进程，由于其紧急程度或重要性的不同，则应提供不同的服务。当患者入院后，除了准备手术的必要时间，其等待第一次手术的时间不能过长，才能更好的保证安排的合理性。

③ 平衡性：

系统中具有多种类型的进程，为了使得系统处于忙碌状态，调度算法应尽可能保持系统资源使用的平衡性。结合眼科病床的安排，医院不能仅

仅治疗一种疾病，而是应该全面的考虑到各种疾病的患者的手术时间和住院时间，平衡地使用医院资源。

#### ④ 策略强制执行：

指在所制定的策略中，只要需要，就必须予以准确地执行，即使造成某些工作的延误也要执行。外伤疾病属于急症，病床有空时应立即安排患者住院，并在住院后第二天安排手术。

#### (2) 批处理系统的目标

##### ① 平均周转时间短：

所谓周转时间，是指从作业被提交给系统开始，到作业完成为止的这段时间间隔（称为作业周转时间）。患者住院时间的长短很大程度上影响了患者的满意度，对于每个患者而言，都希望自己的住院时间最短。

##### ② 系统吞吐量高：

吞吐量是指在单位时间内系统所完成的作业数，它与处理作业的平均长度有关。结合眼科病床的情况，一定时间内的出院人数即反映了医院的吞吐量，如果想要获得高吞吐量，则应尽量多的安排住院时间短的患者。

##### ③ 处理机利用率高：

处理机的利用率一直是衡量系统性能的十分重要的指标，而调度方式和算法又对处理机的利用率起着十分重要的作用。因而合理的病床使用率就是合理的病床安排的重要指标。

因为患者的手术等待时间与患者的入院时间有关，所以患者什么时间手术受到其入院时间的牵制，不同类型的疾病的手术时间需要分情况讨论。而合理的病床安排实际上就是要对白内障手术的时间做出合理的规划。具体对单眼白内障和双眼白内障患者进行分析如下：

对于单眼白内障患者来说，当患者周一入院时需要等到周三才能进行手术，其等待时间为2天；当患者周二入院时周三就可以进行手术，其等待时间为1天；当患者在每周的周三以后（包括周三）入院时，则其只能等到下周的周一才能进行手术。而对于双眼白内障患者来说，因为其手术在周一进行之后需要在周三继续进行另一只眼的手术。当患者在周一入院时需要等到下周一才能进行手术。所以，周六周日尽量安排白内障患者入院，并且尽量安排白内障双眼患者，周一周二尽量安排白内障单眼患者。

具体对患者入院的时间分析如下：

##### (1) 若为周一：

首先安排外伤患者；其次安排单眼白内障，患者准备手术时间为2天；再其次安排其他疾病患者，准备手术时间为2天；最后安排双眼白内障患者入院。

##### (2) 若为周二：

首先安排外伤患者；其次单眼白内障入院，患者等待时间为1天；再其次安排其他疾病患者，准备手术时间为2天；最后安排双眼白内障患者入院。

##### (3) 若为周三、周四或周五：

首先安排外伤患者；其次安排其他疾病患者，术前准备时间为2天；最后安排单双眼白内障患者。

##### (4) 若为周六：

首先安排外伤患者；其次安排双眼白内障，患者准备手术时间为2天；

再其次安排单眼白内障，患者准备手术时间也为 2 天；最后安排其他类型的疾病患者。

(5) 若为周日：

首先安排外伤患者；其次安排双眼白内障，患者准备手术时间为 1 天；再其次安排单眼白内障，患者准备手术时间也为 1 天；最后安排其他类型的疾病患者。

整理各类疾病患者的等待时间得到表 7：

表 3 各类疾病等待时间

|          | 单眼白<br>内障 | 双眼白<br>内障 | 青<br>光眼 | 视网膜<br>疾病 | 外伤 |
|----------|-----------|-----------|---------|-----------|----|
| 周一<br>入院 | 2         | 7         | 3       | 3         | 1  |
| 周二<br>入院 | 1         | 6         | 2       | 2         | 1  |
| 周三<br>入院 | 5         | 5         | 2       | 2         | 1  |
| 周四<br>入院 | 4         | 4         | 2       | 2         | 1  |
| 周五<br>入院 | 3         | 3         | 2       | 2         | 1  |
| 周六<br>入院 | 2         | 2         | 3       | 3         | 1  |
| 周日<br>入院 | 1         | 1         | 2       | 2         | 1  |

接下来我们分析各类疾病的术后观察时间。因为术后观察时间不受手术时间的影响，则我们可以认为对每一类疾病，患者的术后观察时间是相同的。根据实际数据，求得每类疾病的术后观察时间的均值，取最靠近这一均值的整数，得到表 8：

表 4 各类疾病术后观察时间

|            | 单眼白<br>内障 | 双眼<br>白内障 | 青<br>光眼 | 视网膜<br>疾病 | 外<br>伤 |
|------------|-----------|-----------|---------|-----------|--------|
| 术后观察<br>时间 | 3         | 3         | 8       | 10        | 6      |

然后，我们计算各类型疾病的患者入院的优先级，使用 HRRN 模型进行计算。最后利用问题一建立的评价指标体系进行评价。

### 5.3.2 模型的求解

然后，我们使用 HRRN 算法计算各类型疾病的患者入院的优先级。最后利用问题一建立的评价指标体系进行评价。

### 5.2.2 模型的求解

因为服务时间包括术前等待时间和术后观察时间，由表 7 和表 8，我们整理

得到服务时间如表 9 所示：

表 5 各类疾病服务时间

|          | 单眼白<br>内障 | 双眼白<br>内障 | 青<br>光眼 | 视网膜<br>疾病 | 外伤 |
|----------|-----------|-----------|---------|-----------|----|
| 周一<br>入院 | 5         | 10        | 11      | 13        | 7  |
| 周二<br>入院 | 4         | 9         | 10      | 12        | 7  |
| 周三<br>入院 | 8         | 8         | 10      | 12        | 7  |
| 周四<br>入院 | 7         | 7         | 10      | 12        | 7  |
| 周五<br>入院 | 6         | 6         | 10      | 12        | 7  |
| 周六<br>入院 | 5         | 5         | 11      | 13        | 7  |
| 周日<br>入院 | 4         | 4         | 10      | 12        | 7  |

由此可见，对于不同时间不同类型入院的患者，他们的优先级是在不断变化的。

由于 HRRN 模型需要知道某一天有多少个空床位，而 7 月 12 日以前的患者的入院时间无法得知，也就无法确定这些患者的出院时间，而在 8 月 8 日后所有的住院患者的情况都是已知的，因而此后每天的空床位数都是可以计算的，所以我们从 8 月 8 日开始计算。

对于 8 月 8 日以前的患者，保留原来的数据；而对 8 月 8 日以后才入院的患者，保留他们的疾病类型和门诊时间，其他数据使用改进的 HRRN 模型进行计算，即对 8 月 8 日以前入院的患者，他们的出院时间即为题目中给出的出院时间；而对于 8 月 8 日以后入院的患者，他们的出院时间由 HRRN 模型得到。该算法的流程图见图 2：



图 7 算法流程图

使用问题一的评价体系，对我们得到的病床安排模型进行评价，得到的相关结果为综合价值  $S=0.7967$ 。结果比第一问的有所增大，说明我们的模型比医院目前使用的 FCFS 模型的效率更高，并且由对 HRRN 算法的分析可得，此方法可较好的避免 FCFS 算法时间越长队列越长的缺陷。（详细计算结果见附录）

#### 5.4 问题三的建立与求解

为了预测各类型患者入院时间的大致区间，我们决定采用运筹学中的灰色预测模型进行预测并通过计算残差和级比偏差进行检验，得到各类型患者一定时间内每天到达的人数，再通过所得数据得出各类型患者大致的入住区间。

##### 5.4.1 模型的建立

通过分析，我们选用运筹学中的灰色预测模型中的数列预测。灰色模型预测是在复杂的数据中找出一定的规律。其特点是，灰色理论建立的是生成数据模型，不是原始数据模型，即对原始数据作累加生成（或其他方法生成）得到近似的指数规律再进行建模的方法。

我们选用 GM(1, 1) 模型预测是因为 GM(1, 1) 表示的模型是一阶微分方程，且只含一个变量的灰色模型。灰色模型的优点在于不需要很多的数据，精度高；而且运算简便，不考虑分布规律，不考虑变化趋势。



设原始数据序列:

$$x^{(0)} = (x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(14))$$

由一次累加生成序列的定义

$$x^{(1)}(k) = \sum_{i=1}^k x^{(0)}(i), k = 1, 2, \dots, 14$$

对数据进行累加, 得到一次累加生成序列:

$$x^{(1)} = (x^{(1)}(1), x^{(1)}(2), \dots, x^{(1)}(14));$$

同理我们构造 $x^{(1)}$ 的均值生成序列, 如公式?

$$z^{(1)}(k) = \frac{1}{2} (x^{(1)}(k) + x^{(1)}(k-1)), k = 1, 2, \dots, 14$$

$$Z^{(1)} = (Z^{(1)}(2), Z^{(1)}(3), \dots, Z^{(1)}(14));$$

灰色微分方程模型的灰微方程以及其对应的白化方程为公式? :

$$x^{(0)}(k) + ax^{(1)}(k) = b$$

$$\frac{dx^{(1)}}{dt} + ax^{(1)}(t) = b, k = 2, 3, \dots, 14$$

再根据最小二乘法的原理, 对数据进行处理得到白化方程中未知系数  $a$  和  $b$  的估计值:

为了方便对数据进行处理, 我们决定将计算过程全部切换到矩阵的形式:

构造矩阵  $u$ ,  $Y$ ,  $B$  其中  $u = (a, b)^T$ ,

$$Y = (x^{(0)}(2), x^{(0)}(3), \dots, x^{(0)}(14))^T;$$

因此矩阵形式为:

$$B = \begin{pmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \dots & \dots \\ -z^{(1)}(14) & 1 \end{pmatrix}$$

$$Y = Bu$$

由此, 目标转变为最小二乘原理的解决, 即求出一个适合的矩阵  $u$  使得下式达到最小值:

$$J(u) = (Y - Bu)^T(Y - Bu)$$

因此最小二乘  $u$  的估计值为公式?:

$$\hat{u} = (\hat{a}, \hat{b})^T = (B^T B)^{-1} B^T Y$$

把估计值代入求解白化方程得到公式?:

$$\hat{x}^{(1)}(k+1) = \left(x^{(0)}(1) - \frac{\hat{b}}{\hat{a}}\right)e^{-\hat{a}k} + \frac{\hat{b}}{\hat{a}}, k = 0, 1, 2, \dots, 14$$

当  $k=1, 2, \dots$  时，由上式算得的  $\hat{x}^{(1)}(k+1)$  是拟合值；

当  $k \geq N$  时， $\hat{x}^{(1)}(k+1)$  为预报值。

由此我们可以得出预测的累加生成数，依据之前阐述的累加生成数的概念，采用累减的办法，我们就很容易能得到需要预测的值。

但是根据此方法算得的值不能立刻作为结论，需要通过残差检验和级比检验之后才能保证数据的可靠性，这样的预测在现实生活中也才具有实际意义。

#### 5.4.2 模型的求解

(1) 对单眼白内障患者大致入住时间区间的估计：

原始数列：

$$x^{(0)} = (2 \ 1 \ 2 \ 1 \ 3 \ 2 \ 1 \ 1 \ 0 \ 0 \ 4 \ 2 \ 4 \ 1)$$

累加生成列：

$$x^{(1)} = (2 \ 3 \ 5 \ 6 \ 9 \ 11 \ 12 \ 13 \ 13 \ 13 \ 17 \ 19 \ 23 \ 24)$$

均值生成列：

$$z^{(1)} = (2.5 \ 4 \ 5.5 \ 7.5 \ 10 \ 11.5 \ 12.5 \ 13 \ 13 \ 15 \ 18 \ 21 \ 23.5)$$

根据最小二乘法拟合得到：

$$u = (0.0422, 1.1829)$$

根据灰色预测的白化方程即公式??可以得到考研移动端的增长模型满足以下曲线方程：

$$\hat{x}^{(1)}(k+1) = \left(x^{(0)}(1) - \frac{\hat{b}}{\hat{a}}\right)e^{-\hat{a}k} + \frac{\hat{b}}{\hat{a}} = 30.04e^{0.04218t} - 28.04$$

由此我们可以得到：

$$\hat{x}^{(1)}(15) = 26.18$$

由一次累减可以得到：

$$\hat{x}^{(0)}(15) = 2.18$$

因此我们得到在一定时间内的单眼白内障患者的人数每天约为 2 人。

同理我们可以得到双眼白内障、外伤、青光眼和视网膜类疾病患者在一定时

间内每天的人数约为 6 人、2 人、4 人、4 人。整理数据如表 6 所示：

表 6 各类型患者每天到达人数

| 类型          | 单眼<br>白内障 | 双眼<br>白内障 | 外<br>伤 | 青光<br>眼 | 视网膜类<br>疾病 |
|-------------|-----------|-----------|--------|---------|------------|
| 人数(人<br>/天) | 2         | 6         | 2      | 4       | 4          |

最终得到单眼白内障患者的大致入住区间为 1-3 天

同理我们继续进行预测，得双眼白内障、外伤、青光眼和视网膜类疾病患者的大致入住区间分别是 1-4 天、1-2 天、1-4 天、1-4 天。如表 7 所示：

表 7 各类型患者大致入住区间

| 类型   | 单眼白内障 | 双眼白内障 | 外伤    | 青光眼   | 视网膜类疾病 |
|------|-------|-------|-------|-------|--------|
| 入住区间 | 1-3 天 | 1-4 天 | 1-2 天 | 1-4 天 | 1-4 天  |

### 5.4.3 模型的验与分析

(1) 残差检验：

首先定义残差的计算公式如公式 7：

$$\varepsilon(k) = \frac{x^{(0)}(k) - \hat{x}^{(0)}(k)}{x^{(0)}(k)}, k = 1, 2, \dots, 14$$

这里的  $\hat{x}^{(0)}(1) = x^{(0)}(1)$ ，如果  $\varepsilon(k) < 0.1$ ，则可以认为达到一般要求；如果

$\varepsilon(k) \geq 0.1$ ，则可以认为达到较高要求。

(2) 级比偏差值检验：

级比  $(\lambda(k))$  和级比偏差值  $(\rho(k))$  的定义如公式 8：

$$\lambda(k) = \frac{x^{(0)}(k-1)}{x^{(0)}(k)}, k = 1, 2, \dots, 14$$

$$\rho(k) = 1 - \frac{1 - 0.5a}{1 + 0.5a} \lambda(k)$$

一般来说如果  $\rho(k) < 0.1$ ，则认为达到了较高的要求，我们通过计算整理如表 12：（程序见附录三）

表 8 对比级比偏差值的计算

| 类型         | 残差<br>$x^{(0)} - x(0)$ | 级比偏<br>差值 $\rho(k)$ |
|------------|------------------------|---------------------|
| 单眼白内<br>障  | 0                      | 0.0625              |
| 双眼白内<br>障  | 0.0144                 | 0.0304              |
| 外伤         | 0.0608                 | 0.0704              |
| 青光眼        | 0.0344                 | 0.0166              |
| 视网膜类<br>疾病 | 0.0122                 | 0.0205              |

通过对级比偏差值的分析可以得出绝对值最大的级比偏差值为 0.0704，很显然已经达到了最高的要求。所以灰色预测模型的预测值有较高的准确性。

由此我们得出各个类型患者的大致入住时间区间为：单眼白内障患者 5-10 天、双眼白内障患者 7-13 天、外伤患者 1-2 天、青光眼患者 9-15 天、视网膜类疾病患者 8-14 天。整理数据得表 9：

表 9 各类型患者大致入住区间

| 类型        | 单眼白内障  | 双眼白内障  | 外伤    | 青光眼    | 视网膜类<br>疾病 |
|-----------|--------|--------|-------|--------|------------|
| 大致入住时间(天) | 5-10 天 | 7-13 天 | 1-2 天 | 9-15 天 | 8-14 天     |

## 5.5 问题四的模型建立与求解

对手术时间的安排，实际上就是对白内障手术时间的安排，分配好单眼白内障和双眼白内障的手术时间。针对医院周六周日不进行手术，我们将白内障患者的治疗时间分为三类，然后通过问题一计算各个方案的患者满意度，由此得出最合理的手术方案。

### 5.5.1 模型的建立

从医院的角度来说，假如住院部周六、周日不安排手术，那么一段时间内的手术总量会有所下降，从而导致一段时间内治愈的患者总数减少，同时排队的人数会有所增加，导致医院的资源利用率降低。对手术时间的安排，实际上就是对白内障手术时间的安排，由此，我们将手术安排方案基于白内障患者的治疗时间分为三类：周一周三、周二周四、周三周五。以下简称“一三方案”、“二四方案”、“三五方案”。

## 5.5.2 模型的求解

### (1) 对三种方案的分析

通过对白内障手术时间的安排,我们基于白内障患者的治疗时间将手术安排方案分为“一三方案”、“二四方案”、“三五方案”。下面我们分别对这三种方案进行各类型患者术前等待时间和入院优先级的统计分析。

#### ① “一三方案”

即白内障患者在周一周三手术,青光眼和视网膜类疾病患者在周二、周四周五手术。因此各类疾病患者的术前等待时间将会发生变化,重新统计数据得表 14:

表 10 “一三方案”各类型患者术前等待时间

|      | 单眼白内障 | 双眼白内障 | 青光眼 | 视网膜疾病 | 外伤 |
|------|-------|-------|-----|-------|----|
| 周一入院 | 2     | 7     | 3   | 3     | 1  |
| 周二入院 | 1     | 6     | 2   | 2     | 1  |
| 周三入院 | 5     | 5     | 2   | 2     | 1  |
| 周四入院 | 4     | 4     | 5   | 5     | 1  |
| 周五入院 | 3     | 3     | 4   | 4     | 3  |
| 周六入院 | 2     | 2     | 3   | 3     | 2  |
| 周日入院 | 1     | 1     | 2   | 2     | 1  |

由此,我们可以得出各类疾病的优先级也发生了变化。

为了方便统计,我们规定优先级的“最低”、“低”、“中”、“高”、“最高”分别对应数字 1, 2, 3, 4, 5。对于外伤患者而言,因为其属于急症,所以其患者只要入院即为优先级最高的疾病。将“一三方案”中各类疾病患者入院的优先级进行整理得到表 15:

表 11 “一三方案”各类型患者入院优先级

|      | 单眼白内障 | 双眼白内障 | 青光眼 | 视网膜疾病 | 外伤 |
|------|-------|-------|-----|-------|----|
| 周一入院 | 4     | 1     | 3   | 3     | 5  |
| 周二入院 | 4     | 1     | 3   | 3     | 5  |
| 周三入院 | 1     | 1     | 4   | 4     | 5  |
| 周四入院 | 1     | 1     | 1   | 1     | 5  |
| 周五入院 | 1     | 1     | 1   | 1     | 1  |
| 周六入院 | 4     | 5     | 3   | 3     | 1  |
| 周日入院 | 3     | 4     | 1   | 1     | 5  |

#### ② “二四方案”

同理,我们整理“二四方案”各类型患者的入院优先级得表 16:

表 12 “二四方案”各类型患者入院优先级

|      | 单眼白内障 | 双眼白内障 | 青光眼 | 视网膜疾病 | 外伤 |
|------|-------|-------|-----|-------|----|
| 周一入院 | 3     | 4     | 1   | 1     | 5  |
| 周二入院 | 4     | 1     | 3   | 3     | 5  |
| 周三入院 | 4     | 1     | 3   | 3     | 5  |
| 周四入院 | 1     | 1     | 1   | 1     | 5  |
| 周五入院 | 1     | 1     | 1   | 1     | 1  |
| 周六入院 | 1     | 1     | 5   | 5     | 1  |
| 周日入院 | 3     | 4     | 1   | 1     | 5  |

### ③ “三五方案”

整理“三五方案”各类型患者的入院优先级得表 17:

表 13 “三五方案”各类型患者入院优先级

|      | 单眼白内障 | 双眼白内障 | 青光眼 | 视网膜疾病 | 外伤 |
|------|-------|-------|-----|-------|----|
| 周一入院 | 3     | 4     | 1   | 1     | 5  |
| 周二入院 | 3     | 4     | 1   | 1     | 5  |
| 周三入院 | 4     | 1     | 1   | 1     | 5  |
| 周四入院 | 5     | 1     | 1   | 1     | 1  |
| 周五入院 | 1     | 1     | 5   | 5     | 1  |
| 周六入院 | 1     | 1     | 5   | 5     | 1  |
| 周日入院 | 1     | 1     | 4   | 4     | 5  |

(2) 计算问题一中的评价体系

根据问题一我们计算出三种方案的患者满意度分别为  $S_{13} = 0.7882$ ,  $S_{24} = 0.8012$ ,  $S_{35} = 0.8344$ 。因此我们得出, 上述三种方案中选用“三五方案”即在周六周日不安排手术的情况下, 安排白内障手术的时间在周三和周五最为合理。

## 5.6 问题五的模型建立与求解

为了使医院便于管理而采用的病床比例大致固定的方案, 实际上就是当各类患者构成的系统的服务强度相同时, 总服务系统的效率达到最高。因此我们选择建立服务强度平衡模型。通过计算各类型患者的平均到达率和平均服务率, 来计算各类型患者需要的病床数。

### 5.6.1 模型的建立

医院从方便管理的角度, 可以将病床分配比例相对固定, 比例分配合理的情况下, 各类患者会得到有效治疗, 逗留时间会减少, 比例的分配应该根据具体情况来定。当各类患者构成的系统的服务强度相同时, 总的系统服务效率达到最佳。我们分析了数据中各类型患者所占的比例, 得到表 14:

表 14 各类型患者所占百分比

| 类型        | 白内障<br>(单眼) | 白内障<br>(双眼) | 外伤    | 青光眼   | 视网膜<br>类疾病 |
|-----------|-------------|-------------|-------|-------|------------|
| 所占百分比 (%) | 20.6%       | 23.5%       | 15.8% | 11.2% | 28.9%      |

合理的病床分配比例应该在这个比例附近, 在此基础上, 因为要逗留的时间短, 根据最短工作时间原理, 应该将治疗周期短的病人尽量前排。综合分析, 比例的分配应该和某种病的平均治疗周期和所占的比例有关, 因此, 我们选择建立服务强度平衡模型。

五类患者排队系统的参数分别记为:

平均到达率  $\lambda_i$ , 平均服务率  $\mu_i$ , 病床数  $m_i$ ,  $i=1, 2, 3, 4, 5$ ;

系统服务强度:  $\rho_i = \frac{\lambda_i}{m_i \mu_i}$ ,  $i = 1, 2, 3, 4, 5$ ;

总床位数： $m = \sum_{i=1}^5 m_i$ ;

当各系统的服务强度相等时可得

$$m_i = \frac{\lambda_i}{\mu_i \rho_i} = \frac{\frac{\lambda_i}{\mu_i}}{\sum_{j=1}^5 \frac{\lambda_j}{\mu_j}} \times m, i = 1, 2, 3, 4, 5$$

其中， $\lambda_i, \mu_i$ 可以由统计数据得到（具体程序见附录？）。按上述比例得到的各类患者床位比例即为最佳比例，实际操作时，急症病人床位数可以根据需要作变动。

### 5.6.2 模型的求解

根据统计数据，我们可以得到 $\lambda_i, \mu_i$ ，（具体程序见附录？）。

根据公式？、公式？和公式？，将得到的数据整理如表 15 和表 16 所示：

表 15 平均到达率与平均服务率

| 类型 | 平均到达率（人/天） | 平均服务率（人/天） |
|----|------------|------------|
| 1  | 1.639      | 0.189      |
| 2  | 2.180      | 0.118      |
| 3  | 1.059      | 0.143      |
| 4  | 1.033      | 0.095      |
| 5  | 2.787      | 0.079      |

表 16 各类型疾病分配的病床数

| 类型  | 1 | 2  | 3 | 4  | 5  |
|-----|---|----|---|----|----|
| 病床数 | 9 | 18 | 7 | 11 | 34 |

### 5.6.3 模型的检验与分析

由表 16 计算各类型患者所占的比例，与表 14 进行对比得：

表 17 各类型疾病的床数所占比例

| 各类型      | 白内障（单眼） | 白内障（双眼） | 外伤      | 青光眼     | 视网膜类疾病 |
|----------|---------|---------|---------|---------|--------|
| 原始百分比（%） | 20.6 %  | 23.5 %  | 1 5.8 % | 11 .2 % | 28.9 % |
| 所得百分比（%） | 11.4 %  | 22.8 %  | 8 .9 %  | 13 .9 % | 43 %   |
| 偏差（%）    | 9.2 %   | 0.7 %   | 6 .9 %  | 2. 7 %  | 14.1 % |

由此我们可以得出,使得所有病人在系统内的平均逗留时间(含等待入院及住院时间)最短的病床比例分配为单眼白内障病床数为 9 张,双眼白内障病床数为 18 张,外伤病床数为 7 张,青光眼病床数为 11 张,视网膜类疾病病床数为 34 张。

## 六、模型评价与推广

### 6.1 模型的优点

1. 模型对医院的实际问题和统计数据进行了全面分析,同时还可对整个医院的良性运作提供更有价值的评价。

2. 本文对优先级别不同的病症进行了全面系统的床位安排。排队论对于解决医院的床位安排问题准确而合理,不仅解决了医院的床位安排问题还充分的利用了医院的资源。

3. 模型的实施可操作性强,可运用到类似的医院就诊或者床位安排问题中,模型中对病患流的随机性使得该模型更接近实际。

### 6.2 模型的缺点

运用模型解决问题时,仅有两个月的数据,具有一定的局限性,另外对外伤患者都按急症处理,不同病人的术后观察时间也取了平均值,考虑的情况比较简单。

### 6.3 模型的推广

该模型对床位分配问题可扩展运用到就诊,银行等排队模型中,用于解决有资源限制的调度分配问题。

## 参考文献

- [1]言方荣,病床合理安排模型的模糊评价[J],苏教育学院学报(自然科学版),2010,(01):20-23+92.
- [2]杨妍,李康,毛枫雪,谭众,张一,陕勇,院病床优化配置研究[J],东工业技术,2016,(13):299+247.
- [3]赵敬云,葛键宇,陶永东,姜君娜,优先权的动态排队模型安排眼科病床[J],科技资讯,2017,(06):209+211.
- [4]王艳秋,多服务台医院眼科病床排队系统的优化研究[D],天津大学,2014.
- [5]王欣,吴莎莎,简论病床的优化安排[J],河南教育学院学报(自然科学版),2015,(04):29-30.
- [6]安佰玲,徐标,王国华,基于非线性规划的病床安排的优化模型[J],统计与决策,2016,(13):82-85.
- [7]Raheja S, Dhadich R, Rajpal S, Vague Oriented Highest Response Ratio Next (VHRRN) scheduling algorithm[J], 013:1-7.



## 附录

### 附录一：第一问层次分析法 Matlab 代码

```
A=[1,3,3,1/1.5;1/3,1,1,1/4.5;1/3,1,1,1/4.5;3,4.5,4.5,1];
[n,n]=size(A);
x=ones(n,100);
y=ones(n,100);
m=zeros(1,100);
m(1)=max(x(:,1));
y(:,1)=x(:,1);
x(:,2)=A*y(:,1);
m(2)=max(x(:,2));
y(:,2)=x(:,2)/m(2);
p=0.0001;i=2;k=abs(m(2)-m(1));
while k>p
    i=i+1;
    x(:,i)=A*y(:,i-1);
    m(i)=max(x(:,i));
    y(:,i)=x(:,i)/m(i);
    k=abs(m(i)-m(i-1));
end
a=sum(y(:,i));
w=y(:,i)/a;
t=m(i);
disp('权重向量w');disp(w);
disp('最大特征根λ');disp(t);
%一致性检验
CI=(t-n)/(n-1);RI=[0 0 0.52 0.89 1.12 1.26 1.36 1.41 1.46 1.49 1.52 1.54 1.56 1.58 1.59];
CR=CI/RI(n);
if CR<0.10
    disp('通过一致性检验，该矩阵具有满意的一致性');
    disp('CI=');disp(CI);
    disp('CR=');disp(CR);
else
    disp('未通过一致性检验，该矩阵不具有满意的一致性');
end
```

### 附录二：病床安排 C++代码

```
#include <iostream>
#include <string>
#include <fstream>
```

```

#include<stdio.h>
#include<stdlib.h>
#include<iomanip>
using namespace std;
#define M 296
#define N 296
#define S 79
#define D 40
struct JOB//作业结构体
{
    int name; //进程名
    int type; //类型（1 白单，2 白双，3 外伤，4 青光眼，5 视网膜）
    float arrivetime; //到达时间
    float servicetime; //服务时间
    //float waittime; //等待时间
    float starttime; //开始时间
    float finishtime; //完成时间
    float zztime; //周转时间
    float dqzztime; //带权周转时间
    int sfzx; //是否执行标志位（1 已执行，0 未执行）
    float xyb; //记录响应比
};
//记录每张床所剩天数
int waittime;
JOB job_th[M];

//获取下一个要执行的工作
int search_min(JOB job[],int nowtime)
{
    int n;
    //记录符合条件的作业的中间
    JOB job_hrm[M];
    //给中间数组赋值
    for(n=0;n<M;n++)
    {

        if(job[n].arrivetime<=nowtime && job[n].sfzx==0)//若作业已到达未运行
        {

            job_hrm[n].name=job[n].name;
            job_hrm[n].arrivetime=job[n].arrivetime;
            float d;

            if(nowtime%7==6){//若为周四，白单、白双不让入院，外伤一定入院，其他手术不让入院

```

```

switch(job[n].type) {
case 1:
    job_th[n].servicetime=5;//shanchu
    job_hrm[n].xyb=0;
    break;
case 2:
    job_th[n].servicetime=7;//shanchu
    job_hrm[n].xyb=0;
    break;
case 3:
    job_th[n].servicetime=1+6;//准备 1 天，术后观察 6 天
    job_hrm[n].xyb=10;//响应比最高
    break;
case 5:
    job_th[n].servicetime=10;//shanchu
    job_hrm[n].xyb=0;
    break;
case 4:
    job_th[n].servicetime=2+8;//shanchu
    job_hrm[n].xyb=0;
    break;
}

```

}else if(nowtime%7==1){//若为周六，白单、白双让入院，等待时间为 2 天，但是白双优先级高于白单，外伤不能入院，并且优先级最高，其他可入院，准备时间为 3 天

```

switch(job[n].type) {
case 1:
    job_th[n].servicetime=2+3;//准备 2 天，术后观察 3 天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    break;
case 2:
    job_th[n].servicetime=2+3+3;//准备 2 天，术后观察 3 天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    job_hrm[n].xyb+=5;//优先级高于白单小于外伤
    break;
case 3:
    job_th[n].servicetime=7;//shanchu
    job_hrm[n].xyb=0;//响应比最低
    break;
case 5:
    job_th[n].servicetime=3+10;//准备 3 天，术后观察 10 天

```

```

        waittime=nowtime-job_hrn[n].arrivetime;
        d=waittime+job_hrn[n].servicetime;//总服务时间
        job_hrn[n].xyb=d/job_hrn[n].servicetime;//计算高响应比
        break;
    case 4:
        job_th[n].servicetime=3+8;//准备 3 天，术后观察 8 天
        waittime=nowtime-job_hrn[n].arrivetime;
        d=waittime+job_hrn[n].servicetime;//总服务时间
        job_hrn[n].xyb=d/job_hrn[n].servicetime;//计算高响应比
        break;
    }
}
}else if(nowtime%7==2){//若为周日，白单、白双让入院，等待时间为 1 天，但是白双优先级高于白单，外伤一
    定入院，并且优先级最高，其他可入院，准备时间为 2 天
    switch(job[n].type) {
        case 1:
            job_th[n].servicetime=1+3;//准备 1 天，术后观察 3 天
            waittime=nowtime-job_hrn[n].arrivetime;
            d=waittime+job_hrn[n].servicetime;//总服务时间
            job_hrn[n].xyb=d/job_hrn[n].servicetime;//计算高响应比
            break;
        case 2:
            job_th[n].servicetime=1+3+3;//准备 2 天，术中 3 天，术后观察 3 天
            waittime=nowtime-job_hrn[n].arrivetime;
            d=waittime+job_hrn[n].servicetime;//总服务时间
            job_hrn[n].xyb=d/job_hrn[n].servicetime;//计算高响应比
            job_hrn[n].xyb+=5;//优先级高于白单小于外伤
            break;
        case 3:
            job_th[n].servicetime=1+6;//准备 1 天，术后观察 6 天
            job_hrn[n].xyb=10;//响应比最高
            break;
        case 5:
            job_th[n].servicetime=2+10;//准备 2 天，术后观察 10 天
            waittime=nowtime-job_hrn[n].arrivetime;
            d=waittime+job_hrn[n].servicetime;//总服务时间
            job_hrn[n].xyb=d/job_hrn[n].servicetime;//计算高响应比
            break;
        case 4:
            job_th[n].servicetime=2+8;//准备 2 天，术后观察 8 天
            waittime=nowtime-job_hrn[n].arrivetime;
            d=waittime+job_hrn[n].servicetime;//总服务时间
            job_hrn[n].xyb=d/job_hrn[n].servicetime;//计算高响应比
            break;
    }
}

```

```

    }
    if(nowtime%7==3){//若为周一，白单让入院，准备时间为2天；白双不让入院；其他病正常，准备时间为

```

3天

```

switch(job[n].type) {
case 1:
    job_th[n].servicetime=2+3;//准备2天，术后观察3天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    break;
case 2:
    job_th[n].servicetime=7;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 3:
    job_th[n].servicetime=1+6;//准备1天，术后观察6天
    job_hrm[n].xyb=10;//响应比最高
    break;
case 5:
    job_th[n].servicetime=3+10;//准备3天，术后观察10天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    break;
case 4:
    job_th[n].servicetime=3+8;//准备3天，术后观察8天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    break;
}

```

```

} else if(nowtime%7==4){//若为周二，白单让入院，准备时间为1天；白双不让入院；其他病正常，准备时

```

间为2天

```

switch(job[n].type) {
case 1:
    job_th[n].servicetime=1+3;//准备1天，术后观察3天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    break;
case 2:
    job_th[n].servicetime=7;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;

```

```

case 3:
    job_th[n].servicetime=1+6;//准备 1 天，术后观察 6 天
    job_hrn[n].xyb=10;//响应比最高
    break;
case 5:
    job_th[n].servicetime=2+10;//准备 2 天，术后观察 10 天
    waittime=nowtime-job_hrn[n].arrivetime;
    d=waittime+job_hrn[n].servicetime;//总服务时间
    job_hrn[n].xyb=d/job_hrn[n].servicetime;//计算高响应比
    break;
case 4:
    job_th[n].servicetime=2+8;//准备 2 天，术后观察 8 天
    waittime=nowtime-job_hrn[n].arrivetime;
    d=waittime+job_hrn[n].servicetime;//总服务时间
    job_hrn[n].xyb=d/job_hrn[n].servicetime;//计算高响应比
    break;
}
} else if(nowtime==5){//若为周三，白单、白双不让入院，外伤可入，其他可入，准备时间为 2 天
switch(job[n].type) {
case 1:
    job_th[n].servicetime=4;//shanchu
    job_hrn[n].xyb=0;//不让入院
    break;
case 2:
    job_th[n].servicetime=6;//shanchu
    job_hrn[n].xyb=0;//不让入院
    break;
case 3:
    job_th[n].servicetime=1+6;//准备 1 天，术后观察 6 天
    job_hrn[n].xyb=10;//响应比最高
    break;
case 5:
    job_th[n].servicetime=2+10;//准备 2 天，术后观察 10 天
    waittime=nowtime-job_hrn[n].arrivetime;
    d=waittime+job_hrn[n].servicetime;//总服务时间
    job_hrn[n].xyb=d/job_hrn[n].servicetime;//计算高响应比
    break;
case 4:
    job_th[n].servicetime=2+8;//准备 2 天，术后观察 8 天
    waittime=nowtime-job_hrn[n].arrivetime;
    d=waittime+job_hrn[n].servicetime;//总服务时间
    job_hrn[n].xyb=d/job_hrn[n].servicetime;//计算高响应比
    break;
}
}
}

```

```

    }
    else{//若为周五，白单、白双不让入院，外伤不能入院，其他手术不让入院
switch(job[n].type) {
case 1:
    job_th[n].servicetime=5;//shanchu
    job_hrm[n].xyb=0;
    break;
case 2:
    job_th[n].servicetime=7;//shanchu
    job_hrm[n].xyb=0;
    break;
case 3:
    job_th[n].servicetime=7;//shanchu
    job_hrm[n].xyb=0;//响应比最高
    break;
case 5:
    job_th[n].servicetime=10;//shanchu
    job_hrm[n].xyb=0;
    break;
case 4:
    job_th[n].servicetime=2+8;//shanchu
    job_hrm[n].xyb=0;
    break;
}
}

}else//不符合条件赋-1
{
    job_hrm[n].servicetime=-1;
    job_hrm[n].name=-1;
    job_hrm[n].arrivetime=-1;
    job_hrm[n].xyb=-1;
}

}

float temp_xyb;
int temp_name;
//符合条件的作业的服务时间从大到小排序
for(int i=0;i<M;i++)
{
    for(int j=i+1;j<M;j++)
    {
        if(job_hrm[i].xyb<job_hrm[j].xyb)
        {

```

```

        temp_xyb=job_hrrn[i].xyb;
        temp_name=job_hrrn[i].name;
        job_hrrn[i].xyb=job_hrrn[j].xyb;
        job_hrrn[i].name=job_hrrn[j].name;
        job_hrrn[j].xyb=temp_xyb;
        job_hrrn[j].name=temp_name;
    }
}

```

int name;//最小服务时间赋给 num 变量

```

if(job_hrrn[0].xyb!=1)
{
    //num=job_servicetime[k].servicetime;
    name=job_hrrn[0].name;
    //break;
}
else{
    name=-1;
}

```

```

return name;
}

```

int inputnew()//输入作业信息函数

```

{
    ofstream outfile;
    outfile.open("d:\\output13.txt");
    if(outfile==0){
        cout << "Unable to open outfile";
        exit(1);
    }
}

```

int q;

for(q=0;q<M;q++)//循环输入操作

```

{
    job_th[q].name=a[q][0];
    job_th[q].arrivetime=a[q][1];
    job_th[q].type=a[q][2];
}

```



```

        job_th[q].sfzx=0; //标志是否已经执行(未执行)

    }

JOB job_si[M]; //记录七个字段的结构体

//sfzx 字段初值为 0（表示未执行）
for(int s=0;s<M;s++)
{
    job_si[s].sfzx=0;
}

int nowtime=0; //8 月 8 号开始为第 0 天
int l;
int index=0;
while(nowtime<D){
    for(l=0;l<S;l++){
        if(job_th[l].sfzx==0){ //若有空床，安排一个任务
            //获取下一个工作名称
            int name=search_min(job_th,nowtime);
            if(name!=-1){ //按工作名称计算相关值
                for(int f=0;f<M;f++){
                    {
                        if(job_th[f].name==name)
                        {
                            bed[l]+=job_th[f].servicetime;
                            job_si[index].name=job_th[f].name;
                            job_si[index].arrivetime=job_th[f].arrivetime;
                            job_si[index].servicetime=job_th[f].servicetime;
                            job_si[index].starttime=job_si[index].arrivetime+nowtime;
                            job_si[index].finishtime=job_si[index].starttime+job_si[index].servicetime;
                            job_si[index].zztime=job_si[index].finishtime-job_si[index].arrivetime;
                            job_si[index].dqzztime=job_si[index].zztime/job_si[index].servicetime;
                            job_si[index].sfzx=1;
                            job_th[f].sfzx=1;
                            index=index+1;
                            break;
                        }
                    }
                }
            }
        }
    }

    nowtime++;
}

for(l=0;l<S;l++){
    if(bed[l]>0){

```

```

        bed[l]--;
    }
}
nowtime++;
}

//按执行顺序输出工作
cout<<"作业名"<<"\t"<<"到达时间"<<"\t"<<"开始时间"<<"\t"<<"结束时间"<<"\t"<<"周转时间"
"<<"\t"<<"带权周转时间"<<endl;

for(int i=0;i<M;i++)
{
    cout.width(9);
    cout<<job_si[i].name;
    cout.width(9);
    cout<<job_si[i].arrivetime;
    cout.width(11);
    cout<<job_si[i].servicetime;
    cout.width(13);
    cout<<job_si[i].starttime;
    cout.width(10);
    cout<<job_si[i].finishtime;
    cout.width(9);
    cout<<job_si[i].zztime;
    cout.width(9);
    cout<<job_si[i].dqzztime<<endl;

    //写入文件

    outfile<<job_si[i].name<<"\t"<<job_si[i].arrivetime<<"\t"<<job_si[i].starttime<<"\t"<<job_si[i].finishtime<<"\t"<<job_si[i].zztime<<en
    dl;

}

//获取周转时间和平均周转时间
double sum_zz=0;
double sum_dqzz=0;
for(int p=0;p<M;p++)
{
    sum_zz=sum_zz+job_si[p].zztime;
    sum_dqzz=sum_dqzz+job_si[p].dqzztime;
}
cout<<"平均周转时间: "<<sum_zz/M<<endl;

outfile.close();

return 0;
};

```

```
int main(int argc, char* argv[])
{
    inputnew();
    return 0;
}
```

### 附录三：灰色预测 Matlab 代码

```
clc,clear
x0=[2 5 3 5 1 5 5 1 1 2 3 1 1 5]';
n=length(x0);%n=6
lamda=x0(1:n-1)./x0(2:n); %1/4E Ë ã 4 ¶ ± È
range=minmax(lamda); %1/4E Ë ã 4 ¶ ± È ¶ ï §
x1=cumsum(x0) %À Û %Ó Ô Ë Ë ã
B=[-0.5*(x1(1:n-1)+x1(2:n)),ones(n-1,1)];
Y=x0(2:n);
u=B\Y;%Ã ã ¶ Ì Ë ÿ
syms x(t)
x=dsolve(diff(x)+u(1)*x==u(2),x(0)==x0(1)) %Ç ó Ì ¢ Ö ½ ¶ ì Ä · ù %½ ã
y=vpa(x,4) %Æ ã Ö Ð ì Ä ± Ë ã Ì Ö Ë 4 Ì Ë ÿ × Ö
yuce1=subs(x,t,[0:n-1]) %Ç ó Ö Ñ Ö ¢ ÿ 4 Ý Ö ¶ ã Ö ì
yuce1=double(yuce1)
yuce=[x0(1),diff(yuce1)] % ¶ Ì Ö Ö Ë Ë ã » Ö Ë ÿ 4 Ý
%1/4E Ë ã Ð ¶
epsilon=x0'-yuce
delta=abs(epsilon./x0') %1/4E Ë ã Ì ã ¶ Ö Ì ó ¶
rho=1-(1-0.5*u(1))/(1+0.5*u(1))*lamda' %1/4E Ë ã 4 ¶ ± È Ë « ? Ö ì u(1)=a
```

### 附录四：问题四“一三组合”c++代码

```
#include <iostream>
#include <string>
#include <fstream>
#include<stdio.h>
#include<stdlib.h>
#include<iomanip>
using namespace std;
#define M 296
#define N 296
#define S 79
#define D 40
struct JOB//作业结构体
{
    int name; //进程名
```

```

int type;//类型（1 白单，2 白双，3 外伤，4 青光眼，5 视网膜）
float arrivetime;//到达时间
float servicetime;//服务时间
//float waittime;//等待时间
float starttime; //开始时间
float finishtime;//完成时间
float zztime; //周转时间
float dqzztime; //带权周转时间
int sfzx;//是否执行标志位（1 已执行，0 未执行）
float xyb;//记录响应比
};
int waittime;
JOB job_th[M];

//获取下一个要执行的工作
int search_min(JOB job[],int nowtime)
{
    int n;
    //记录符合条件的作业的中间
    JOB job_hrm[M];
    //给中间数组赋值
    for(n=0;n<M;n++)
    {

        if(job[n].arrivetime<=nowtime && job[n].sfzx==0)//若作业已到达未运行
        {

            job_hrm[n].name=job[n].name;
            job_hrm[n].arrivetime=job[n].arrivetime;
            float d;

            if(nowtime%7==6){//若为周四，白单、白双不让入院，外伤一定入院，其他手术不让入院
                switch(job[n].type) {
                    case 1:
                        job_th[n].servicetime=5;//shanchu
                        job_hrm[n].xyb=0;
                        break;
                    case 2:
                        job_th[n].servicetime=7;//shanchu
                        job_hrm[n].xyb=0;
                        break;
                    case 3:
                        job_th[n].servicetime=1+6;//准备1天，术后观察6天
                        job_hrm[n].xyb=10;//响应比最高

```

```

        break;
    case 5:
        job_th[n].servicetime=10;//shanchu
        job_hrm[n].xyb=0;
        break;
    case 4:
        job_th[n].servicetime=2+8;//shanchu
        job_hrm[n].xyb=0;
        break;
    }
}
} else if(nowtime%7==1){//若为周六，白单、白双让入院，等待时间为2天，但是白双优先级高于白单，外伤
不能入院，并且优先级最高，其他可入院，准备时间为3天
    switch(job[n].type) {
        case 1:
            job_th[n].servicetime=2+3;//准备2天，术后观察3天
            waittime=nowtime-job_hrm[n].arrivetime;
            d=waittime+job_hrm[n].servicetime;//总服务时间
            job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
            break;
        case 2:
            job_th[n].servicetime=2+3+3;//准备2天，术后观察3天
            waittime=nowtime-job_hrm[n].arrivetime;
            d=waittime+job_hrm[n].servicetime;//总服务时间
            job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
            job_hrm[n].xyb+=5;//优先级高于白单小于外伤
            break;
        case 3:
            job_th[n].servicetime=7;//shanchu
            job_hrm[n].xyb=0;//响应比最低
            break;
        case 5:
            job_th[n].servicetime=3+10;//准备3天，术后观察10天
            waittime=nowtime-job_hrm[n].arrivetime;
            d=waittime+job_hrm[n].servicetime;//总服务时间
            job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
            break;
        case 4:
            job_th[n].servicetime=3+8;//准备3天，术后观察8天
            waittime=nowtime-job_hrm[n].arrivetime;
            d=waittime+job_hrm[n].servicetime;//总服务时间
            job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
            break;
    }
}
} else if(nowtime%7==2){//若为周日，白单、白双让入院，等待时间为1天，但是白双优先级高于白单，外伤一

```

定入院，并且优先级最高，其他可入院，准备时间为 2 天

```
switch(job[n].type) {
    case 1:
        job_th[n].servicetime=1+3;//准备 1 天，术后观察 3 天
        waittime=nowtime-job_hrm[n].arrivetime;
        d=waittime+job_hrm[n].servicetime;//总服务时间
        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
        break;
    case 2:
        job_th[n].servicetime=1+3+3;//准备 2 天，术中 3 天，术后观察 3 天
        waittime=nowtime-job_hrm[n].arrivetime;
        d=waittime+job_hrm[n].servicetime;//总服务时间
        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
        job_hrm[n].xyb+=5;//优先级高于白单小于外伤
        break;
    case 3:
        job_th[n].servicetime=1+6;//准备 1 天，术后观察 6 天
        job_hrm[n].xyb=10;//响应比最高
        break;
    case 5:
        job_th[n].servicetime=2+10;//准备 2 天，术后观察 10 天
        waittime=nowtime-job_hrm[n].arrivetime;
        d=waittime+job_hrm[n].servicetime;//总服务时间
        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
        break;
    case 4:
        job_th[n].servicetime=2+8;//准备 2 天，术后观察 8 天
        waittime=nowtime-job_hrm[n].arrivetime;
        d=waittime+job_hrm[n].servicetime;//总服务时间
        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
        break;
}

if(nowtime%7==3){//若为周一，白单让入院，准备时间为 2 天；白双不让入院；其他病正常，准备时间为
```

3 天

```
switch(job[n].type) {
    case 1:
        job_th[n].servicetime=2+3;//准备 2 天，术后观察 3 天
        waittime=nowtime-job_hrm[n].arrivetime;
        d=waittime+job_hrm[n].servicetime;//总服务时间
        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
        break;
    case 2:
        job_th[n].servicetime=7;//shanchu
```

```

        job_hrm[n].xyb=0;//不让入院
        break;
    case 3:
        job_th[n].servicetime=1+6;//准备 1 天，术后观察 6 天
        job_hrm[n].xyb=10;//响应比最高
        break;
    case 5:
        job_th[n].servicetime=3+10;//准备 3 天，术后观察 10 天
        waittime=nowtime-job_hrm[n].arrivetime;
        d=waittime+job_hrm[n].servicetime;//总服务时间
        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
        break;
    case 4:
        job_th[n].servicetime=3+8;//准备 3 天，术后观察 8 天
        waittime=nowtime-job_hrm[n].arrivetime;
        d=waittime+job_hrm[n].servicetime;//总服务时间
        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
        break;
    }
} else if(nowtime%7==4){//若为周二，白单让入院，准备时间为 1 天；白双不让入院；其他病正常，准备时

```

间为 2 天

```

switch(job[n].type) {
    case 1:
        job_th[n].servicetime=1+3;//准备 1 天，术后观察 3 天
        waittime=nowtime-job_hrm[n].arrivetime;
        d=waittime+job_hrm[n].servicetime;//总服务时间
        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
        break;
    case 2:
        job_th[n].servicetime=7;//shanchu
        job_hrm[n].xyb=0;//不让入院
        break;
    case 3:
        job_th[n].servicetime=1+6;//准备 1 天，术后观察 6 天
        job_hrm[n].xyb=10;//响应比最高
        break;
    case 5:
        job_th[n].servicetime=2+10;//准备 2 天，术后观察 10 天
        waittime=nowtime-job_hrm[n].arrivetime;
        d=waittime+job_hrm[n].servicetime;//总服务时间
        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
        break;
    case 4:
        job_th[n].servicetime=2+8;//准备 2 天，术后观察 8 天

```

```

        waittime=nowtime-job_hrm[n].arrivetime;
        d=waittime+job_hrm[n].servicetime;//总服务时间
        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
        break;
    }
} else if(nowtime==5){//若为周三，白单、白双不让入院，外伤可入，其他可入，准备时间为2天
switch(job[n].type) {
case 1:
    job_th[n].servicetime=4;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 2:
    job_th[n].servicetime=6;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 3:
    job_th[n].servicetime=1+6;//准备1天，术后观察6天
    job_hrm[n].xyb=10;//响应比最高
    break;
case 5:
    job_th[n].servicetime=2+10;//准备2天，术后观察10天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    break;
case 4:
    job_th[n].servicetime=2+8;//准备2天，术后观察8天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    break;
}
}
else{//若为周五，白单、白双不让入院，外伤不能入院，其他手术不让入院
switch(job[n].type) {
case 1:
    job_th[n].servicetime=5;//shanchu
    job_hrm[n].xyb=0;
    break;
case 2:
    job_th[n].servicetime=7;//shanchu
    job_hrm[n].xyb=0;
    break;
case 3:

```



```

        job_th[n].servicetime=7;//shanchu
        job_hrrn[n].xyb=0;//响应比最高
        break;
    case 5:
        job_th[n].servicetime=10;//shanchu
        job_hrrn[n].xyb=0;
        break;
    case 4:
        job_th[n].servicetime=2+8;//shanchu
        job_hrrn[n].xyb=0;
        break;
    }
    }
} else //不符合条件赋-1
{
    job_hrrn[n].servicetime=-1;
    job_hrrn[n].name=-1;
    job_hrrn[n].arrivetime=-1;
    job_hrrn[n].xyb=-1;
}
}

```

```

float temp_xyb;
int temp_name;
//符合条件的作业的服务时间从大到小排序
for(int i=0;i<M;i++)
{
    for(int j=i+1;j<M;j++)
    {
        if(job_hrrn[i].xyb<job_hrrn[j].xyb)
        {
            temp_xyb=job_hrrn[i].xyb;
            temp_name=job_hrrn[i].name;
            job_hrrn[i].xyb=job_hrrn[j].xyb;
            job_hrrn[i].name=job_hrrn[j].name;
            job_hrrn[j].xyb=temp_xyb;
            job_hrrn[j].name=temp_name;
        }
    }
}

```

```

int name;//最小服务时间赋给 num 变量

```

```

if(job_hrrn[0].xyb!=-1)
{
    //num=job_servicetime[k].servicetime;
    name=job_hrrn[0].name;

    //break;
}
else{
    name=-1;
}

return name;
}

```

```

int inputnew()//输入作业信息函数
{
    ofstream outfile;
    outfile.open("d:\\output13.txt");
    if(outfile==0){
        cout << "Unable to open outfile";
        exit(1);
    }
}

```

```

int q;

```

```

for(q=0;q<M;q++)//循环输入操作
{
    job_th[q].name=a[q][0];
    job_th[q].arrivetime=a[q][1];
    job_th[q].type=a[q][2];
    job_th[q].sfzx=0; //标志是否已经执行(未执行)

}

```

```

JOB job_si[M];//记录七个字段的结构体

```

```

//sfzx 字段初值为 0（表示未执行）
for(int s=0;s<M;s++)
{
    job_si[s].sfzx=0;
}

```

```

int nowtime=0;//8 月 8 号开始为第 0 天

int l;

int index=0;

while(nowtime<D){
    for(l=0;l<S;l++){
        if.bed[l]==0){//若有空床，安排一个任务

            //获取下一个工作名称

            int name=search_min(job_th,nowtime);
            if(name!=-1){//按工作名称计算相关值
                for(int f=0;f<M;f++){
                    {
                        if(job_th[f].name==name)
                        {
                            bed[l]+=job_th[f].servicetime;
                            job_si[index].name=job_th[f].name;
                            job_si[index].arrivetime=job_th[f].arrivetime;
                            job_si[index].servicetime=job_th[f].servicetime;
                            job_si[index].starttime=job_si[index].arrivetime+nowtime;
                            job_si[index].finishtime=job_si[index].starttime+job_si[index].servicetime;
                            job_si[index].zztime=job_si[index].finishtime-job_si[index].arrivetime;
                            job_si[index].dqzztime=job_si[index].zztime/job_si[index].servicetime;
                            job_si[index].sfzx=1;
                            job_th[f].sfzx=1;
                            index=index+1;

                            break;
                        }
                    }
                }
            }

        }

        for(l=0;l<S;l++){
            if.bed[l]>0){
                bed[l]--;
            }
        }

        nowtime++;
    }

    //按执行顺序输出工作

    cout<<"作业名"<<"\t"<<"到达时间"<<"\t"<<"开始时间"<<"\t"<<"结束时间"<<"\t"<<"周转时间"
    "<<"\t"<<"带权周转时间"<<endl;

    for(int i=0;i<M;i++)
    {
        cout.width(9);

```

```

        cout<<job_si[i].name;

        cout.width(9);

        cout<<job_si[i].arrivetime;

        cout.width(11);

        cout<<job_si[i].servicetime;

        cout.width(13);

        cout<<job_si[i].starttime;

        cout.width(10);

        cout<<job_si[i].finishtime;

        cout.width(9);

        cout<<job_si[i].zztime;

        cout.width(9);

        cout<<job_si[i].dqzztime<<endl;

        //写入文件

        outfile<<job_si[i].name<<"t"<<job_si[i].arrivetime<<"t"<<job_si[i].starttime<<"t"<<job_si[i].finishtime<<"t"<<job_si[i].zztime<<endl;

    }

    //获取周转时间和平均周转时间

    double sum_zz=0;

    double sum_dqzz=0;

    for(int p=0;p<M;p++)

    {

        sum_zz=sum_zz+job_si[p].zztime;

        sum_dqzz=sum_dqzz+job_si[p].dqzztime;

    }

    cout<<"平均周转时间: "<<sum_zz/M<<endl;

}

outfile.close();

return 0;

};

int main(int argc, char* argv[])

{

    inputnew();

    return 0;

}

```

## 附录五：问题四“二四组合”c++代码

```

#include <iostream>

#include <string>

#include <fstream>

#include<stdio.h>

```

```

#include<stdlib.h>
#include<iomanip>
using namespace std;
#define M 296
#define N 296
#define S 79
#define D 40
struct JOB//作业结构体
{
    int name; //进程名
    int type; //类型（1 白单，2 白双，3 外伤，4 青光眼，5 视网膜）
    float arrivetime; //到达时间
    float servicetime; //服务时间
    //float waittime; //等待时间
    float starttime; //开始时间
    float finishtime; //完成时间
    float zztime; //周转时间
    float dqzztime; //带权周转时间
    int sfzx; //是否执行标志位（1 已执行，0 未执行）
    float xyb; //记录响应比
};
int waittime;
JOB job_th[M];
//获取下一个要执行的工作
int search_min(JOB job[], int nowtime)
{
    int n;
    //记录符合条件的作业的中间
    JOB job_hrm[M];
    //给中间数组赋值
    for(n=0; n<M; n++)
    {
        if(job[n].arrivetime<=nowtime && job[n].sfzx==0) //若作业已到达未运行
        {
            job_hrm[n].name=job[n].name;
            job_hrm[n].arrivetime=job[n].arrivetime;
            float d;

            if(nowtime%7==6) { //若为周四，白单、白双不让入院，外伤一定入院，其他手术不让入院
                switch(job[n].type) {
                    case 1:
                        job_th[n].servicetime=5; //shanchu
                        job_hrm[n].xyb=0;

```

```

        break;
case 2:
    job_th[n].servicetime=7;//shanchu
    job_hrrn[n].xyb=0;
    break;
case 3:
    job_th[n].servicetime=1+6;//准备 1 天，术后观察 6 天
    job_hrrn[n].xyb=10;//响应比最高
    break;
case 5:
    job_th[n].servicetime=10;//shanchu
    job_hrrn[n].xyb=0;
    break;
case 4:
    job_th[n].servicetime=2+8;//shanchu
    job_hrrn[n].xyb=0;
    break;
    }
} else if(nowtime%7==2){//若为周日，白单、白双让入院，等待时间为 2 天，但是白双优先级高于白单，外伤
一定入院，并且优先级最高，其他可入院，准备时间为 2 天
switch(job[n].type) {
    case 1:
        job_th[n].servicetime=2+3;//准备 2 天，术后观察 3 天
        waittime=nowtime-job_hrrn[n].arrivetime;
        d=waittime+job_hrrn[n].servicetime;//总服务时间
        job_hrrn[n].xyb=d/job_hrrn[n].servicetime;//计算高响应比
        break;
    case 2:
        job_th[n].servicetime=2+3+3;//准备 2 天，术后观察 3 天
        waittime=nowtime-job_hrrn[n].arrivetime;
        d=waittime+job_hrrn[n].servicetime;//总服务时间
        job_hrrn[n].xyb=d/job_hrrn[n].servicetime;//计算高响应比
        job_hrrn[n].xyb+=5;//优先级高于白单小于外伤
        break;
    case 3:
        job_th[n].servicetime=1+6;//准备 1 天，术后观察 6 天
        job_hrrn[n].xyb=10;//响应比最高
        break;
    case 5:
        job_th[n].servicetime=3+10;//准备 3 天，术后观察 10 天
        waittime=nowtime-job_hrrn[n].arrivetime;
        d=waittime+job_hrrn[n].servicetime;//总服务时间
        job_hrrn[n].xyb=d/job_hrrn[n].servicetime;//计算高响应比
        break;

```

```

        case 4:
            job_th[n].servicetime=3+8;//准备 3 天，术后观察 8 天
            waittime=nowtime-job_hrm[n].arrivetime;
            d=waittime+job_hrm[n].servicetime;//总服务时间
            job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
            break;
    }
}
} else if(nowtime%7==3){//若为周一，白单、白双让入院，等待时间为 1 天，但是白双优先级高于白单，外伤一
    定入院，并且优先级最高，其他可入院，准备时间为 2 天
    switch(job[n].type) {
        case 1:
            job_th[n].servicetime=1+3;//准备 1 天，术后观察 3 天
            waittime=nowtime-job_hrm[n].arrivetime;
            d=waittime+job_hrm[n].servicetime;//总服务时间
            job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
            break;
        case 2:
            job_th[n].servicetime=1+3+3;//准备 2 天，术中 3 天，术后观察 3 天
            waittime=nowtime-job_hrm[n].arrivetime;
            d=waittime+job_hrm[n].servicetime;//总服务时间
            job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
            job_hrm[n].xyb+=5;//优先级高于白单小于外伤
            break;
        case 3:
            job_th[n].servicetime=1+6;//准备 1 天，术后观察 6 天
            job_hrm[n].xyb=10;//响应比最高
            break;
        case 5:
            job_th[n].servicetime=2+10;//准备 2 天，术后观察 10 天
            waittime=nowtime-job_hrm[n].arrivetime;
            d=waittime+job_hrm[n].servicetime;//总服务时间
            job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
            break;
        case 4:
            job_th[n].servicetime=2+8;//准备 2 天，术后观察 8 天
            waittime=nowtime-job_hrm[n].arrivetime;
            d=waittime+job_hrm[n].servicetime;//总服务时间
            job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
            break;
    }
}
}
if(nowtime%7==4){//若为周二，白单让入院，准备时间为 2 天；白双不让入院；其他病正常，准备时间为
    3 天
    switch(job[n].type) {

```

```

case 1:
    job_th[n].servicetime=2+3;//准备 2 天，术后观察 3 天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    break;
case 2:
    job_th[n].servicetime=7;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 3:
    job_th[n].servicetime=1+6;//准备 1 天，术后观察 6 天
    job_hrm[n].xyb=10;//响应比最高
    break;
case 5:
    job_th[n].servicetime=3+10;//准备 3 天，术后观察 10 天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    break;
case 4:
    job_th[n].servicetime=3+8;//准备 3 天，术后观察 8 天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    break;
}
} else if(nowtime%7==5){//若为周三，白单让入院，准备时间为 1 天；白双不让入院；其他病正常，准备时

```

间为 2 天

```

switch(job[n].type) {
case 1:
    job_th[n].servicetime=1+3;//准备 1 天，术后观察 3 天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    break;
case 2:
    job_th[n].servicetime=7;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 3:
    job_th[n].servicetime=1+6;//准备 1 天，术后观察 6 天
    job_hrm[n].xyb=10;//响应比最高
    break;

```



```

case 5:
    job_th[n].servicetime=2+10;//准备 2 天，术后观察 10 天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    break;
case 4:
    job_th[n].servicetime=2+8;//准备 2 天，术后观察 8 天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    break;
}
} else if(nowtime%7==0){//若为周五，全部不可入
switch(job[n].type) {
case 1:
    job_th[n].servicetime=4;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 2:
    job_th[n].servicetime=6;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 3:
    job_th[n].servicetime=7;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 5:
    job_th[n].servicetime=8;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 4:
    job_th[n].servicetime=10;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
}
} else{//若为周六，白单、白双不让入院，外伤不让入院，其他可入院，准备时间为两天
switch(job[n].type) {
case 1:
    job_th[n].servicetime=4;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 2:
    job_th[n].servicetime=6;//shanchu

```

```

        job_hrm[n].xyb=0;//不让入院
        break;
    case 3:
        job_th[n].servicetime=7;//准备 1 天，术后观察 6 天
        job_hrm[n].xyb=0;//响应比最高
        break;
    case 5:
        job_th[n].servicetime=2+10;//准备 2 天，术后观察 10 天
        waittime=nowtime-job_hrm[n].arrivetime;
        d=waittime+job_hrm[n].servicetime;//总服务时间
        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
        break;
    case 4:
        job_th[n].servicetime=2+8;//准备 2 天，术后观察 8 天
        waittime=nowtime-job_hrm[n].arrivetime;
        d=waittime+job_hrm[n].servicetime;//总服务时间
        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
        break;
    }
} else//不符合条件赋-1
{
    job_hrm[n].servicetime=-1;
    job_hrm[n].name=-1;
    job_hrm[n].arrivetime=-1;
    job_hrm[n].xyb=-1;
}
}

float temp_xyb;
int temp_name;
//符合条件的作业的服务时间从大到小排序
for(int i=0;i<M;i++)
{
    for(int j=i+1;j<M;j++)
    {
        if(job_hrm[i].xyb<job_hrm[j].xyb)
        {
            temp_xyb=job_hrm[i].xyb;
            temp_name=job_hrm[i].name;
            job_hrm[i].xyb=job_hrm[j].xyb;
            job_hrm[i].name=job_hrm[j].name;
            job_hrm[j].xyb=temp_xyb;

```

```

        job_hrrn[j].name=temp_name;
    }
}

```

```

int name;//最小服务时间赋给 num 变量

```

```

if(job_hrrn[0].xyb!=-1)
{
    //num=job_servicetime[k].servicetime;
    name=job_hrrn[0].name;

    //break;
}
else{
    name=-1;
}
return name;
}

```

```

int inputnew();//输入作业信息函数

```

```

{
    ofstream outfile;
    outfile.open("d:\\output24.txt");
    if(outfile==0){
        cout << "Unable to open outfile";
        exit(1);
    }
}

```

```

int q;

```

```

for(q=0;q<M;q++)//循环输入操作

```

```

{
    job_th[q].name=a[q][0];
    job_th[q].arrivetime=a[q][1];
    job_th[q].type=a[q][2];
    job_th[q].sfzx=0; //标志是否已经执行(未执行)

}

```

```

JOB job_si[M];//记录七个字段的结构体

```

```

//sfzx 字段初值为 0（表示未执行）
for(int s=0;s<M;s++)
{
    job_si[s].sfzx=0;
}

int nowtime=0;//8 月 8 号开始为第 0 天
int l;
int index=0;
while(nowtime<D){
    for(l=0;l<S;l++){
        if(job_th[l].sfzx==0){//若有空床，安排一个任务
            //获取下一个工作名称
            int name=search_min(job_th,nowtime);
            if(name!=-1){//按工作名称计算相关值
                for(int f=0;f<M;f++){
                    {
                        if(job_th[f].name==name)
                        {
                            bed[l]+=job_th[f].servicetime;
                            job_si[index].name=job_th[f].name;
                            job_si[index].arrivetime=job_th[f].arrivetime;
                            job_si[index].servicetime=job_th[f].servicetime;
                            job_si[index].starttime=job_si[index].arrivetime+nowtime;
                            job_si[index].finishtime=job_si[index].starttime+job_si[index].servicetime;
                            job_si[index].zztime=job_si[index].finishtime-job_si[index].arrivetime;
                            job_si[index].dqzztime=job_si[index].zztime/job_si[index].servicetime;
                            job_si[index].sfzx=1;
                            job_th[f].sfzx=1;
                            index=index+1;
                            break;
                        }
                    }
                }
            }
        }
    }
    nowtime++;
}

```

```

//按执行顺序输出工作
cout.width(9);
cout<<"作业名";

    cout.width(9);
cout<<"到达时间";

    cout.width(11);
cout<<"服务时间";

    cout.width(13);
cout<<"开始时间";

    cout.width(10);
cout<<"结束时间";

    cout.width(10);
cout<<"周转时间";

    cout.width(9);
cout<<"带权周转时间"<<endl;

for(int i=0;i<M;i++)
{
    cout.width(9);
    cout<<job_si[i].name;

    cout.width(9);
    cout<<job_si[i].arrivetime;

    cout.width(11);
    cout<<job_si[i].servicetime;

    cout.width(13);
    cout<<job_si[i].starttime;

    cout.width(10);
    cout<<job_si[i].finishtime;

    cout.width(9);
    cout<<job_si[i].zztime;

    cout.width(9);
    cout<<job_si[i].dqzztime<<endl;

    //写入文件

    outfile<<job_si[i].name<<"t"<<job_si[i].arrivetime<<"t"<<job_si[i].starttime<<"t"<<job_si[i].finishtime<<"t"<<job_si[i].zztime<<endl;

}

//获取周转时间和平均周转时间
double sum_zz=0;
double sum_dqzz=0;
for(int p=0;p<M;p++)
{
    sum_zz=sum_zz+job_si[p].zztime;
    sum_dqzz=sum_dqzz+job_si[p].dqzztime;
}

```

```

    }

    cout<<"平均周转时间: "<<sum_zz/M<<endl;

    //cout<<"平均带权周转时间: "<<sum_dqzz/M<<endl;

    outfile.close();

    return 0;
};

int main(int argc, char* argv[])
{
    inputnew();
    return 0;
}

```

#### 附录六：问题四“三五组合”c++代码

```

#include <iostream>
#include <string>
#include <fstream>
#include <stdio.h>
#include <stdlib.h>
#include <iomanip>
using namespace std;

#define M 296
#define N 296
#define S 79
#define D 40

struct JOB//作业结构体
{
    int name; //进程名
    int type; //类型（1 白单，2 白双，3 外伤，4 青光眼，5 视网膜）
    float arrivetime; //到达时间
    float servicetime; //服务时间
    //float waittime; //等待时间
    float starttime; //开始时间
    float finishtime; //完成时间
    float zztime; //周转时间
    float dqzztime; //带权周转时间
    int sfzx; //是否执行标志位（1 已执行，0 未执行）
    float xyb; //记录响应比
};

int waittime;
JOB job_th[M];

//获取下一个要执行的工作
int search_min(JOB job[],int nowtime)

```

```

{
    int n;
    //记录符合条件的作业的中间
    JOB job_hrm[M];
    //给中间数组赋值
    int d;
    for(n=0;n<M;n++)
    {
        if(job[n].arrivetime<=nowtime && job[n].sfzx==0)//若作业已到达未运行
        {
            job_hrm[n].name=job[n].name;
            job_hrm[n].arrivetime=job[n].arrivetime;
            float d;
            if(nowtime%7==2){//若为周日，白单、白双不让入院，外伤一定入院，其他手术可入，准备时间为两天
                switch(job[n].type) {
                    case 1:
                        job_th[n].servicetime=5;//shanchu
                        job_hrm[n].xyb=0;
                        break;
                    case 2:
                        job_th[n].servicetime=7;//shanchu
                        job_hrm[n].xyb=0;
                        break;
                    case 3:
                        job_th[n].servicetime=1+6;//准备 1 天，术后观察 6 天
                        job_hrm[n].xyb=10;//响应比最高
                        break;
                    case 5:
                        job_th[n].servicetime=2+10;//准备 2 天，术后观察 10 天
                        waittime=nowtime-job_hrm[n].arrivetime;
                        d=waittime+job_hrm[n].servicetime;//总服务时间
                        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
                        break;
                    case 4:
                        job_th[n].servicetime=2+8;//准备 2 天，术后观察 8 天
                        waittime=nowtime-job_hrm[n].arrivetime;
                        d=waittime+job_hrm[n].servicetime;//总服务时间
                        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
                        break;
                }
            }else if(nowtime%7==3){//若为周一，白单、白双让入院，等待时间为 2 天，但是白双优先级高于白单，外伤一定入院，并且优先级最高，其他可入院，准备时间为 3 天
                switch(job[n].type) {
                    case 1:

```

```

job_th[n].servicetime=2+3;//准备 2 天，术后观察 3 天
waittime=nowtime-job_hrm[n].arrivetime;
d=waittime+job_hrm[n].servicetime;//总服务时间
job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
break;

```

case 2:

```

job_th[n].servicetime=2+3+3;//准备 2 天，术后观察 3 天
waittime=nowtime-job_hrm[n].arrivetime;
d=waittime+job_hrm[n].servicetime;//总服务时间
job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
job_hrm[n].xyb+=5;//优先级高于白单小于外伤
break;

```

case 3:

```

job_th[n].servicetime=1+6;//准备 1 天，术后观察 6 天
job_hrm[n].xyb=10;//响应比最高
break;

```

case 5:

```

job_th[n].servicetime=3+10;//准备 3 天，术后观察 10 天
waittime=nowtime-job_hrm[n].arrivetime;
d=waittime+job_hrm[n].servicetime;//总服务时间
job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
break;

```

case 4:

```

job_th[n].servicetime=3+8;//准备 3 天，术后观察 8 天
waittime=nowtime-job_hrm[n].arrivetime;
d=waittime+job_hrm[n].servicetime;//总服务时间
job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
break;

```

```

}

```

}else if(nowtime%7==4){//若为周二，白单、白双让入院，等待时间为 1 天，但是白双优先级高于白单，外伤一定入院，并且优先级最高，其他可入院，准备时间为 2 天

```

switch(job[n].type) {

```

case 1:

```

job_th[n].servicetime=1+3;//准备 1 天，术后观察 3 天
waittime=nowtime-job_hrm[n].arrivetime;
d=waittime+job_hrm[n].servicetime;//总服务时间
job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
break;

```

case 2:

```

job_th[n].servicetime=1+3+3;//准备 2 天，术中 3 天，术后观察 3 天
waittime=nowtime-job_hrm[n].arrivetime;
d=waittime+job_hrm[n].servicetime;//总服务时间
job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
job_hrm[n].xyb+=5;//优先级高于白单小于外伤

```



```

        break;
    case 3:
        job_th[n].servicetime=1+6;//准备 1 天，术后观察 6 天
        job_hrm[n].xyb=10;//响应比最高
        break;
    case 5:
        job_th[n].servicetime=2+10;//准备 2 天，术后观察 10 天
        waittime=nowtime-job_hrm[n].arrivetime;
        d=waittime+job_hrm[n].servicetime;//总服务时间
        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
        break;
    case 4:
        job_th[n].servicetime=2+8;//准备 2 天，术后观察 8 天
        waittime=nowtime-job_hrm[n].arrivetime;
        d=waittime+job_hrm[n].servicetime;//总服务时间
        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
        break;
    }
    }else if(nowtime%7==5){//若为周三，白单让入院，准备时间为 2 天；白双不让入院；外伤可入，其他病正
    不让入院
    switch(job[n].type) {
    case 1:
        job_th[n].servicetime=2+3;//准备 2 天，术后观察 3 天
        waittime=nowtime-job_hrm[n].arrivetime;
        d=waittime+job_hrm[n].servicetime;//总服务时间
        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
        break;
    case 2:
        job_th[n].servicetime=7;//shanchu
        job_hrm[n].xyb=0;//不让入院
        break;
    case 3:
        job_th[n].servicetime=1+6;//准备 1 天，术后观察 6 天
        job_hrm[n].xyb=10;//响应比最高
        break;
    case 5:
        job_th[n].servicetime=8;//shanchu
        job_hrm[n].xyb=0;//不让入院
        break;
    case 4:
        job_th[n].servicetime=10;//shanchu
        job_hrm[n].xyb=0;//不让入院
        break;
    }
}

```

```

} else if(nowtime%7==6){//若为周四，白单让入院，准备时间为1天；白双不让入院；其他病不让入院
switch(job[n].type) {
case 1:
    job_th[n].servicetime=1+3;//准备1天，术后观察3天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    break;
case 2:
    job_th[n].servicetime=7;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 3:
    job_th[n].servicetime=1+6;//准备1天，术后观察6天
    job_hrm[n].xyb=10;//响应比最高
    break;
case 5:
    job_th[n].servicetime=8;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 4:
    job_th[n].servicetime=10;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
}
} else if(nowtime%7==0){//若为周五，白不可，外伤不可，其他可入，准备时间为3天
switch(job[n].type) {
case 1:
    job_th[n].servicetime=4;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 2:
    job_th[n].servicetime=6;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 3:
    job_th[n].servicetime=7;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 5:
    job_th[n].servicetime=3+10;//准备3天，术后观察10天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比

```

```

        break;
    case 4:
        job_th[n].servicetime=3+8;//准备 3 天，术后观察 8 天
        waittime=nowtime-job_hrm[n].arrivetime;
        d=waittime+job_hrm[n].servicetime;//总服务时间
        job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
        break;
    }
}
}
else{//若为周六，白单、白双不让入院，外伤不让入院，其他可入院，准备时间为两天
switch(job[n].type) {
case 1:
    job_th[n].servicetime=4;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 2:
    job_th[n].servicetime=6;//shanchu
    job_hrm[n].xyb=0;//不让入院
    break;
case 3:
    job_th[n].servicetime=7;//准备 1 天，术后观察 6 天
    job_hrm[n].xyb=0;//响应比最高
    break;
case 5:
    job_th[n].servicetime=2+10;//准备 2 天，术后观察 10 天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    break;
case 4:
    job_th[n].servicetime=2+8;//准备 2 天，术后观察 8 天
    waittime=nowtime-job_hrm[n].arrivetime;
    d=waittime+job_hrm[n].servicetime;//总服务时间
    job_hrm[n].xyb=d/job_hrm[n].servicetime;//计算高响应比
    break;
}
}
}
}
else//不符合条件赋-1
{
    job_hrm[n].servicetime=-1;
    job_hrm[n].name=-1;
    job_hrm[n].arrivetime=-1;
    job_hrm[n].xyb=-1;
}
}
}

```

```

    }

    float temp_xyb;
    int temp_name;
    //符合条件的作业的服务时间从大到小排序
    for(int i=0;i<M;i++)
    {
        for(int j=i+1;j<M;j++)
        {
            if(job_hrrn[i].xyb<job_hrrn[j].xyb)
            {
                temp_xyb=job_hrrn[i].xyb;
                temp_name=job_hrrn[i].name;
                job_hrrn[i].xyb=job_hrrn[j].xyb;
                job_hrrn[i].name=job_hrrn[j].name;
                job_hrrn[j].xyb=temp_xyb;
                job_hrrn[j].name=temp_name;
            }
        }
    }

    int name;//最小服务时间赋给 num 变量

    if(job_hrrn[0].xyb!=1)
    {
        //num=job_servicetime[k].servicetime;
        name=job_hrrn[0].name;

        //break;
    }
    else{
        name=-1;
    }

    return name;
}

```

```

int inputnew()//输入作业信息函数
{
    ofstream outfile;
    outfile.open("d:\\output35.txt");
    if(outfile==0){
        cout << "Unable to open outfile";
    }
}

```

```

        exit(1);
    }

    int q;

    for(q=0;q<M;q++)//循环输入操作
    {
        job_th[q].name=a[q][0];
        job_th[q].arrivetime=a[q][1];
        job_th[q].type=a[q][2];
        job_th[q].sfzx=0; //标志是否已经执行(未执行)
    }

    JOB job_si[M]; //记录七个字段的结构体

    //sfzx 字段初值为 0 (表示未执行)
    for(int s=0;s<M;s++)
    {
        job_si[s].sfzx=0;
    }

    int nowtime=0; //8 月 8 号开始为第 0 天
    int l;
    int index=0;
    while(nowtime<D){
        for(l=0;l<S;l++){
            if(job_th[l].sfzx==0){ //若有空床，安排一个任务
                //获取下一个工作名称
                int name=search_min(job_th,nowtime);
                if(name!=-1){ //按工作名称计算相关值
                    for(int f=0;f<M;f++){
                        {
                            if(job_th[f].name==name)
                            {
                                bed[l]+=job_th[f].servicetime;
                                job_si[index].name=job_th[f].name;
                                job_si[index].arrivetime=job_th[f].arrivetime;
                                job_si[index].servicetime=job_th[f].servicetime;
                                job_si[index].starttime=job_si[index].arrivetime+nowtime;
                                job_si[index].finishtime=job_si[index].starttime+job_si[index].servicetime;
                                job_si[index].zztime=job_si[index].finishtime-job_si[index].arrivetime;
                                job_si[index].dqzztime=job_si[index].zztime/job_si[index].servicetime;

```

```

        job_si[index].sfzx=1;

        job_th[f].sfzx=1;

        index=index+1;

        break;

    }

}

}

}

for(l=0;l<S;l++){
    if(bed[l]>0){
        bed[l]--;
    }
}

nowtime++;
}

```

//按执行顺序输出工作

```

cout.width(9);

cout<<"作业名";

    cout.width(9);

cout<<"到达时间";

    cout.width(11);

cout<<"服务时间";

    cout.width(13);

cout<<"开始时间";

    cout.width(10);

cout<<"结束时间";

    cout.width(10);

cout<<"周转时间";

    cout.width(9);

cout<<"带权周转时间"<<endl;

```

```

for(int i=0;i<M;i++)
{
    cout.width(9);

    cout<<job_si[i].name;

    cout.width(9);

    cout<<job_si[i].arrivetime;

    cout.width(11);

    cout<<job_si[i].servicetime;

    cout.width(13);

    cout<<job_si[i].starttime;

    cout.width(10);

    cout<<job_si[i].finishtime;
}

```

```

        cout.width(9);
        cout<<job_si[i].zztime;
        cout.width(9);
        cout<<job_si[i].dqzztime<<endl;
        //写入文件
        outfile<<job_si[i].name<<"t"<<job_si[i].arrivetime<<"t"<<job_si[i].starttime<<"t"<<job_si[i].finishtime<<"t"<<job_si[i].zztime<<endl;
    }
    //获取周转时间和平均周转时间
    double sum_zz=0;
    double sum_dqzz=0;
    for(int p=0;p<M;p++)
    {
        sum_zz=sum_zz+job_si[p].zztime;
        sum_dqzz=sum_dqzz+job_si[p].dqzztime;
    }
    cout<<"平均周转时间: "<<sum_zz/M<<endl;
    //cout<<"平均带权周转时间: "<<sum_dqzz/M<<endl;

    outfile.close();

    return 0;
};

int main(int argc, char* argv[])
{
    inputnew();
    return 0;
}

```

### 附录七：问题五计算各病所占床数 Matlab 代码

```

clc,clear

t=0;
x=[1.64,2.18,1.06,1.033,2.787];
y=[0.189,0.118,0.143,0.095,0.079];
for j=1:5
    t=t+x(j)/y(j);
end
for j=1:5
    m(j)=(x(j)/y(j))/t*79
end

```

附录八：根据该住院部当前的情况拟出院病人人数

| 序号 | 类型      | 门诊时间      | 入院时间      | 第一次手术时间   | 第二次手术时间   | 出院时间      |
|----|---------|-----------|-----------|-----------|-----------|-----------|
| 1  | 视网膜疾病   | 2008-8-15 | 2008-8-29 | 2008-8-31 | /         | 2008-9-12 |
| 2  | 视网膜疾病   | 2008-8-16 | 2008-8-29 | 2008-8-31 | /         | 2008-9-12 |
| 3  | 白内障(双眼) | 2008-8-19 | 2008-9-1  | 2008-9-8  | 2008-9-10 | 2008-9-13 |
| 4  | 青光眼     | 2008-8-19 | 2008-9-1  | 2008-9-4  | /         | 2008-9-12 |
| 5  | 视网膜疾病   | 2008-8-19 | 2008-9-1  | 2008-9-4  | /         | 2008-9-14 |
| 6  | 视网膜疾病   | 2008-8-19 | 2008-9-1  | 2008-9-4  | /         | 2008-9-14 |
| 7  | 白内障(双眼) | 2008-8-19 | 2008-9-1  | 2008-9-8  | 2008-9-10 | 2008-9-13 |
| 8  | 视网膜疾病   | 2008-8-19 | 2008-9-2  | 2008-9-4  | /         | 2008-9-14 |
| 9  | 视网膜疾病   | 2008-8-19 | 2008-9-3  | 2008-9-5  | /         | 2008-9-15 |
| 10 | 白内障(双眼) | 2008-8-19 | 2008-9-3  | 2008-9-8  | 2008-9-10 | 2008-9-13 |
| 11 | 白内障(双眼) | 2008-8-19 | 2008-9-3  | 2008-9-8  | 2008-9-10 | 2008-9-13 |
| 12 | 视网膜疾病   | 2008-8-19 | 2008-9-3  | 2008-9-5  | /         | 2008-9-15 |
| 13 | 白内障     | 2008-8-19 | 2008-9-4  | 2008-9-8  | /         | 2008-9-12 |
| 14 | 视网膜疾病   | 2008-8-19 | 2008-9-4  | 2008-9-6  | /         | 2008-9-16 |
| 15 | 视网膜疾病   | 2008-8-20 | 2008-9-4  | 2008-9-6  | /         | 2008-9-16 |
| 16 | 视网膜疾病   | 2008-8-20 | 2008-9-4  | 2008-9-6  | /         | 2008-9-16 |
| 17 | 视网膜疾病   | 2008-8-20 | 2008-9-4  | 2008-9-6  | /         | 2008-9-16 |
| 18 | 视网膜疾病   | 2008-8-20 | 2008-9-4  | 2008-9-6  | /         | 2008-9-16 |
| 19 | 白内障(双眼) | 2008-8-20 | 2008-9-4  | 2008-9-8  | 2008-9-10 | 2008-9-13 |
| 20 | 视网膜疾病   | 2008-8-21 | 2008-9-5  | 2008-9-7  | /         | 2008-9-17 |
| 21 | 白内障(双眼) | 2008-8-22 | 2008-9-5  | 2008-9-8  | 2008-9-10 | 2008-9-13 |
| 22 | 白内障(双眼) | 2008-8-22 | 2008-9-5  | 2008-9-8  | 2008-9-10 | 2008-9-13 |
| 23 | 视网膜疾病   | 2008-8-22 | 2008-9-5  | 2008-9-7  | /         | 2008-9-17 |
| 24 | 青光眼     | 2008-8-23 | 2008-9-5  | 2008-9-7  | /         | 2008-9-15 |
| 25 | 青光眼     | 2008-8-23 | 2008-9-5  | 2008-9-7  | /         | 2008-9-15 |
| 26 | 白内障(双眼) | 2008-8-23 | 2008-9-5  | 2008-9-8  | 2008-9-10 | 2008-9-13 |
| 27 | 视网膜疾病   | 2008-8-23 | 2008-9-6  | 2008-9-9  | /         | 2008-9-19 |
| 28 | 白内障(双眼) | 2008-8-23 | 2008-9-6  | 2008-9-8  | 2008-9-10 | 2008-9-13 |
| 29 | 白内障(双眼) | 2008-8-23 | 2008-9-6  | 2008-9-8  | 2008-9-10 | 2008-9-13 |
| 30 | 青光眼     | 2008-8-24 | 2008-9-6  | 2008-9-9  | /         | 2008-9-16 |
| 31 | 视网膜疾病   | 2008-8-24 | 2008-9-6  | 2008-9-9  | /         | 2008-9-16 |
| 32 | 白内障(双眼) | 2008-8-24 | 2008-9-6  | 2008-9-8  | 2008-9-10 | 2008-9-13 |
| 33 | 视网膜疾病   | 2008-8-24 | 2008-9-6  | 2008-9-9  | /         | 2008-9-19 |
| 34 | 视网膜疾病   | 2008-8-24 | 2008-9-6  | 2008-9-9  | /         | 2008-9-19 |
| 35 | 白内障(双眼) | 2008-8-24 | 2008-9-6  | 2008-9-8  | 2008-9-10 | 2008-9-13 |
| 36 | 青光眼     | 2008-8-24 | 2008-9-6  | 2008-9-9  | /         | 2008-9-16 |
| 37 | 视网膜疾病   | 2008-8-25 | 2008-9-6  | 2008-9-9  | /         | 2008-9-19 |
| 38 | 视网膜疾病   | 2008-8-25 | 2008-9-6  | 2008-9-9  | /         | 2008-9-19 |
| 39 | 青光眼     | 2008-8-25 | 2008-9-6  | 2008-9-9  | /         | 2008-9-16 |
| 40 | 白内障(双眼) | 2008-8-25 | 2008-9-6  | 2008-9-8  | 2008-9-10 | 2008-9-13 |
| 41 | 白内障(双眼) | 2008-8-25 | 2008-9-7  | 2008-9-8  | 2008-9-10 | 2008-9-13 |



|    |         |           |           |           |           |           |
|----|---------|-----------|-----------|-----------|-----------|-----------|
| 42 | 视网膜疾病   | 2008-8-26 | 2008-9-7  | 2008-9-9  | /         | 2008-9-19 |
| 43 | 白内障     | 2008-8-26 | 2008-9-7  | 2008-9-8  | /         | 2008-9-12 |
| 44 | 白内障(双眼) | 2008-8-26 | 2008-9-7  | 2008-9-8  | 2008-9-10 | 2008-9-13 |
| 45 | 视网膜疾病   | 2008-8-26 | 2008-9-7  | 2008-9-9  | /         | 2008-9-19 |
| 46 | 视网膜疾病   | 2008-8-27 | 2008-9-8  | 2008-9-11 | /         | 2008-9-21 |
| 47 | 视网膜疾病   | 2008-8-27 | 2008-9-8  | 2008-9-11 | /         | 2008-9-21 |
| 48 | 白内障     | 2008-8-27 | 2008-9-8  | 2008-9-10 | /         | 2008-9-13 |
| 49 | 视网膜疾病   | 2008-8-27 | 2008-9-8  | 2008-9-11 | /         | 2008-9-21 |
| 50 | 白内障(双眼) | 2008-8-27 | 2008-9-9  | 2008-9-15 | 2008-9-17 | 2008-9-20 |
| 51 | 视网膜疾病   | 2008-8-28 | 2008-9-9  | 2008-9-11 | /         | 2008-9-21 |
| 52 | 白内障     | 2008-8-28 | 2008-9-9  | 2008-9-10 | /         | 2008-9-13 |
| 53 | 白内障(双眼) | 2008-8-28 | 2008-9-10 | 2008-9-15 | 2008-9-17 | 2008-9-20 |
| 54 | 视网膜疾病   | 2008-8-28 | 2008-9-10 | 2008-9-12 | /         | 2008-9-22 |
| 55 | 视网膜疾病   | 2008-8-28 | 2008-9-10 | 2008-9-12 | /         | 2008-9-22 |
| 56 | 视网膜疾病   | 2008-8-28 | 2008-9-10 | 2008-9-12 | /         | 2008-9-22 |
| 57 | 白内障(双眼) | 2008-8-28 | 2008-9-10 | 2008-9-15 | 2008-9-17 | 2008-9-20 |
| 58 | 白内障(双眼) | 2008-8-28 | 2008-9-10 | 2008-9-15 | 2008-9-17 | 2008-9-20 |
| 59 | 青光眼     | 2008-8-28 | 2008-9-9  | 2008-9-11 | /         | 2008-9-19 |
| 60 | 青光眼     | 2008-8-29 | 2008-9-10 | 2008-9-12 | /         | 2008-9-20 |
| 61 | 视网膜疾病   | 2008-8-29 | 2008-9-10 | 2008-9-12 | /         | 2008-9-22 |
| 62 | 青光眼     | 2008-8-29 | 2008-9-10 | 2008-9-12 | /         | 2008-9-20 |
| 63 | 白内障(双眼) | 2008-8-29 | 2008-9-10 | 2008-9-15 | 2008-9-17 | 2008-9-20 |
| 64 | 视网膜疾病   | 2008-8-29 | 2008-9-10 | 2008-9-12 | /         | 2008-9-22 |
| 65 | 白内障     | 2008-8-29 | 2008-9-11 | 2008-9-15 | /         | 2008-9-18 |
| 66 | 白内障(双眼) | 2008-8-29 | 2008-9-11 | 2008-9-15 | 2008-9-17 | 2008-9-20 |
| 67 | 白内障     | 2008-8-29 | 2008-9-11 | 2008-9-15 | /         | 2008-9-18 |
| 68 | 白内障(双眼) | 2008-8-29 | 2008-9-11 | 2008-9-15 | 2008-9-17 | 2008-9-20 |
| 69 | 视网膜疾病   | 2008-8-30 | 2008-9-11 | 2008-9-13 | /         | 2008-9-23 |
| 70 | 白内障     | 2008-8-30 | 2008-9-11 | 2008-9-15 | /         | 2008-9-18 |
| 71 | 视网膜疾病   | 2008-8-30 | 2008-9-11 | 2008-9-13 | /         | 2008-9-23 |
| 72 | 外伤      | 2008-9-4  | 2008-9-5  | 2008-9-6  | /         | 2008-9-12 |
| 73 | 外伤      | 2008-9-5  | 2008-9-6  | 2008-9-7  | /         | 2008-9-13 |
| 74 | 外伤      | 2008-9-5  | 2008-9-6  | 2008-9-7  | /         | 2008-9-13 |
| 75 | 外伤      | 2008-9-5  | 2008-9-6  | 2008-9-7  | /         | 2008-9-13 |
| 76 | 外伤      | 2008-9-6  | 2008-9-7  | 2008-9-8  | /         | 2008-9-14 |
| 77 | 外伤      | 2008-9-8  | 2008-9-9  | 2008-9-10 | /         | 2008-9-16 |
| 78 | 外伤      | 2008-9-9  | 2008-9-10 | 2008-9-11 | /         | 2008-9-17 |
| 79 | 外伤      | 2008-9-9  | 2008-9-10 | 2008-9-11 | /         | 2008-9-17 |

## 附录九：问题二对该住院部当前情况的病人安排方案

| 序号 | 类型      | 门诊时间      | 入院时间      | 第一次手术时间   | 第二次手术时间   | 出院时间      |
|----|---------|-----------|-----------|-----------|-----------|-----------|
| 1  | 白内障(双眼) | 2008/8/30 | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 2  | 视网膜疾病   | 2008/8/30 | 2008/9/12 | 2008/9/14 | /         | 2008/9/24 |
| 3  | 青光眼     | 2008/8/30 | 2008/9/12 | 2008/9/14 | /         | 2008/9/21 |
| 4  | 视网膜疾病   | 2008/8/30 | 2008/9/12 | 2008/9/14 | /         | 2008/9/24 |
| 5  | 视网膜疾病   | 2008/8/30 | 2008/9/12 | 2008/9/14 | /         | 2008/9/24 |
| 6  | 白内障(双眼) | 2008/8/30 | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 7  | 白内障     | 2008/8/31 | 2008/9/16 | 2008/9/17 | /         | 2008/9/20 |
| 8  | 青光眼     | 2008/8/31 | 2008/9/12 | 2008/9/14 | /         | 2008/9/21 |
| 9  | 白内障(双眼) | 2008/8/31 | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 10 | 视网膜疾病   | 2008/8/31 | 2008/9/15 | 2008/9/18 | /         | 2008/9/28 |
| 11 | 视网膜疾病   | 2008/8/31 | 2008/9/15 | 2008/9/18 | /         | 2008/9/28 |
| 12 | 视网膜疾病   | 2008/8/31 | 2008/9/17 | 2008/9/19 | /         | 2008/9/29 |
| 13 | 青光眼     | 2008/8/31 | 2008/9/17 | 2008/9/19 | /         | 2008/9/26 |
| 14 | 白内障     | 2008/8/31 | 2008/9/16 | 2008/9/17 | /         | 2008/9/20 |
| 15 | 视网膜疾病   | 2008/9/1  | 2008/9/17 | 2008/9/19 | /         | 2008/9/29 |
| 16 | 视网膜疾病   | 2008/9/1  | 2008/9/17 | 2008/9/19 | /         | 2008/9/29 |
| 17 | 青光眼     | 2008/9/1  | 2008/9/18 | 2008/9/20 | /         | 2008/9/27 |
| 18 | 白内障(双眼) | 2008/9/1  | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 19 | 白内障(双眼) | 2008/9/1  | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 20 | 白内障(双眼) | 2008/9/1  | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 21 | 视网膜疾病   | 2008/9/1  | 2008/9/18 | 2008/9/20 | /         | 2008/9/30 |
| 22 | 白内障     | 2008/9/1  | 2008/9/16 | 2008/9/17 | /         | 2008/9/20 |
| 23 | 视网膜疾病   | 2008/9/1  | 2008/9/18 | 2008/9/20 | /         | 2008/9/30 |
| 24 | 视网膜疾病   | 2008/9/1  | 2008/9/18 | 2008/9/20 | /         | 2008/9/30 |
| 25 | 白内障     | 2008/9/2  | 2008/9/16 | 2008/9/17 | /         | 2008/9/20 |
| 26 | 白内障     | 2008/9/2  | 2008/9/16 | 2008/9/17 | /         | 2008/9/20 |

|    |         |          |           |           |           |           |
|----|---------|----------|-----------|-----------|-----------|-----------|
| 27 | 白内障(双眼) | 2008/9/2 | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 28 | 白内障     | 2008/9/2 | 2008/9/16 | 2008/9/17 | /         | 2008/9/20 |
| 29 | 视网膜疾病   | 2008/9/2 | 2008/9/19 | 2008/9/21 | /         | 2008-9-31 |
| 30 | 视网膜疾病   | 2008/9/3 | 2008/9/19 | 2008/9/21 | /         | 2008-9-31 |
| 31 | 视网膜疾病   | 2008/9/3 | 2008/9/19 | 2008/9/21 | /         | 2008-9-31 |
| 32 | 白内障(双眼) | 2008/9/3 | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 33 | 白内障     | 2008/9/3 | 2008/9/16 | 2008/9/17 | /         | 2008/9/20 |
| 34 | 视网膜疾病   | 2008/9/3 | 2008/9/19 | 2008/9/21 | /         | 2008-9-31 |
| 35 | 白内障     | 2008/9/3 | 2008/9/16 | 2008/9/17 | /         | 2008/9/20 |
| 36 | 视网膜疾病   | 2008/9/3 | 2008/9/19 | 2008/9/21 | /         | 2008-9-31 |
| 37 | 视网膜疾病   | 2008/9/3 | 2008/9/19 | 2008/9/21 | /         | 2008-9-31 |
| 38 | 白内障(双眼) | 2008/9/4 | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 39 | 白内障     | 2008/9/4 | 2008/9/16 | 2008/9/17 | /         | 2008/9/20 |
| 40 | 青光眼     | 2008/9/4 | 2008/9/19 | 2008/9/21 | /         | 2008/9/28 |
| 41 | 视网膜疾病   | 2008/9/4 | 2008/9/19 | 2008/9/21 | /         | 2008-9-31 |
| 42 | 视网膜疾病   | 2008/9/4 | 2008/9/19 | 2008/9/21 | /         | 2008-9-31 |
| 43 | 视网膜疾病   | 2008/9/4 | 2008/9/19 | 2008/9/21 | /         | 2008-9-31 |
| 44 | 青光眼     | 2008/9/4 | 2008/9/19 | 2008/9/21 | /         | 2008/9/28 |
| 45 | 白内障(双眼) | 2008/9/4 | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 46 | 白内障(双眼) | 2008/9/4 | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 47 | 青光眼     | 2008/9/4 | 2008/9/20 | 2008/9/23 | /         | 2008/9/30 |
| 48 | 青光眼     | 2008/9/4 | 2008/9/20 | 2008/9/23 | /         | 2008/9/30 |
| 49 | 视网膜疾病   | 2008/9/4 | 2008/9/20 | 2008/9/23 | /         | 2008/10/2 |
| 50 | 视网膜疾病   | 2008/9/4 | 2008/9/20 | 2008/9/23 | /         | 2008/10/2 |
| 51 | 白内障(双眼) | 2008/9/5 | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 52 | 白内障(双眼) | 2008/9/5 | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 53 | 白内障(双眼) | 2008/9/5 | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 54 | 视网膜疾病   | 2008/9/5 | 2008/9/20 | 2008/9/23 | /         | 2008/10/2 |

|    |         |           |           |           |           |           |
|----|---------|-----------|-----------|-----------|-----------|-----------|
| 55 | 白内障(双眼) | 2008/9/5  | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 56 | 青光眼     | 2008/9/5  | 2008/9/20 | 2008/9/23 | /         | 2008/9/30 |
| 57 | 白内障(双眼) | 2008/9/5  | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 58 | 白内障     | 2008/9/5  | 2008/9/20 | 2008/9/22 | /         | 2008/9/25 |
| 59 | 白内障(双眼) | 2008/9/5  | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 60 | 白内障(双眼) | 2008/9/5  | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 61 | 白内障(双眼) | 2008/9/6  | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 62 | 视网膜疾病   | 2008/9/6  | 2008/9/20 | 2008/9/23 | /         | 2008/10/2 |
| 63 | 青光眼     | 2008/9/6  | 2008/9/20 | 2008/9/23 | /         | 2008/9/30 |
| 64 | 白内障(双眼) | 2008/9/6  | 2008/9/13 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 65 | 视网膜疾病   | 2008/9/7  | 2008/9/20 | 2008/9/23 | /         | 2008/10/2 |
| 66 | 白内障(双眼) | 2008/9/7  | 2008/9/14 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 67 | 视网膜疾病   | 2008/9/7  | 2008/9/20 | 2008/9/23 | /         | 2008/10/2 |
| 68 | 白内障     | 2008/9/8  | 2008/9/20 | 2008/9/22 | /         | 2008/9/25 |
| 69 | 视网膜疾病   | 2008/9/8  | 2008/9/20 | 2008/9/23 | /         | 2008/10/2 |
| 70 | 视网膜疾病   | 2008/9/8  | 2008/9/20 | 2008/9/23 | /         | 2008/10/2 |
| 71 | 白内障     | 2008/9/8  | 2008/9/20 | 2008/9/22 | /         | 2008/9/25 |
| 72 | 白内障(双眼) | 2008/9/8  | 2008/9/14 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 73 | 白内障     | 2008/9/8  | 2008/9/20 | 2008/9/22 | /         | 2008/9/25 |
| 74 | 视网膜疾病   | 2008/9/8  | 2008/9/20 | 2008/9/22 | /         | 2008/10/2 |
| 75 | 白内障     | 2008/9/8  | 2008/9/20 | 2008/9/23 | /         | 2008/9/25 |
| 76 | 青光眼     | 2008/9/9  | 2008/9/20 | 2008/9/23 | /         | 2008/9/30 |
| 77 | 青光眼     | 2008/9/9  | 2008/9/20 | 2008/9/23 | /         | 2008/9/30 |
| 78 | 视网膜疾病   | 2008/9/9  | 2008/9/20 | 2008/9/23 | /         | 2008/10/2 |
| 79 | 白内障     | 2008/9/9  | 2008/9/20 | 2008/9/22 | /         | 2008/9/25 |
| 80 | 白内障     | 2008/9/9  | 2008/9/20 | 2008/9/22 | /         | 2008/9/25 |
| 81 | 视网膜疾病   | 2008/9/10 | 2008/9/20 | 2008/9/23 | /         | 2008/10/2 |
| 82 | 白内障     | 2008/9/10 | 2008/9/20 | 2008/9/22 | /         | 2008/9/25 |
| 83 | 白内障(双眼) | 2008/9/10 | 2008/9/14 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 84 | 白内障     | 2008/9/10 | 2008/9/20 | 2008/9/22 | /         | 2008/9/25 |

|     |         |           |           |           |           |           |
|-----|---------|-----------|-----------|-----------|-----------|-----------|
| 85  | 白内障     | 2008/9/10 | 2008/9/20 | 2008/9/22 | /         | 2008/9/25 |
| 86  | 白内障(双眼) | 2008/9/10 | 2008/9/14 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 87  | 白内障     | 2008/9/10 | 2008/9/20 | 2008/9/22 | /         | 2008/9/25 |
| 88  | 青光眼     | 2008/9/10 | 2008/9/20 | 2008/9/23 | /         | 2008/9/30 |
| 89  | 白内障(双眼) | 2008/9/10 | 2008/9/14 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 90  | 视网膜疾病   | 2008/9/11 | 2008/9/20 | 2008/9/23 | /         | 2008/10/2 |
| 91  | 视网膜疾病   | 2008/9/11 | 2008/9/20 | 2008/9/23 | /         | 2008/10/2 |
| 92  | 青光眼     | 2008/9/11 | 2008/9/20 | 2008/9/23 | /         | 2008/9/30 |
| 93  | 白内障(双眼) | 2008/9/11 | 2008/9/14 | 2008/9/15 | 2008/9/17 | 2008/9/20 |
| 94  | 白内障(双眼) | 2008/9/11 | 2008/9/20 | 2008/9/22 | 2008/9/24 | 2008/9/27 |
| 95  | 青光眼     | 2008/9/11 | 2008/9/20 | 2008/9/23 | /         | 2008/9/30 |
| 96  | 白内障(双眼) | 2008/9/11 | 2008/9/20 | 2008/9/22 | 2008/9/24 | 2008/9/27 |
| 97  | 外伤      | 2008/9/11 | 2008/9/12 | 2008/9/13 | /         | 2008/9/19 |
| 98  | 白内障(双眼) | 2008/9/11 | 2008/9/20 | 2008/9/22 | 2008/9/24 | 2008/9/27 |
| 99  | 视网膜疾病   | 2008/9/11 | 2008/9/20 | 2008/9/23 | /         | 2008/10/2 |
| 100 | 白内障     | 2008/9/11 | 2008/9/20 | 2008/9/22 | /         | 2008/9/25 |
| 101 | 视网膜疾病   | 2008/9/11 | 2008/9/20 | 2008/9/23 | /         | 2008/10/2 |
| 102 | 视网膜疾病   | 2008/9/11 | 2008/9/20 | 2008/9/23 | /         | 2008/10/2 |