

**IE206 Project Report**

**2021-2022 Spring**

***Scheduling for a Local Hospital***

*Academic integrity is expected of all students of METU at all times, whether in the presence or absence of members of the faculty.*

*Understanding this, we declare that we shall not give, use, or receive unauthorized aid in this project.*

*Student ID Number Full Name Signature*

**2444974 Cafer Selli** 

**2587848 Zeynep Beril Şahin** metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Table of Contents

[Introduction 1](#_Toc107695530)

[Objective 1 1](#_Toc107695531)

[Flowchart of the Algorithm 1](#_Toc107695532)

[Objective 2 5](#_Toc107695533)

[Flowchart of the Algorithm 5](#_Toc107695534)

[Conclusion 7](#_Toc107695535)

Introduction

Solving scheduling problems of institutions enables decision-makers to develop and implement more efficient policies that directly serve the needs and expectations. By implementing these policies, resources are used in the most efficient and purposeful way.

This project embraces two different heuristics to solve the scheduling problem of operations in a hospital with a given number of rooms and a given time interval. For the given two different objective functions, which may differ in real life, the heuristics are transformed into algorithms to adjust the scheduling of operations expediently.

Objective 1

In this objective, our purpose is maximizing the number of operations which is completed within its initial available time interval. In order to fulfill this objective, the designed algorithm consider each person one by one and arrange them to the interval where there is minimum number of collisions occur. In terms of collision, how many of the patients are available at that time interval is considered. By counting the number of the patients available at that time we calculated the collision matrix. Using this calculation, a new matrix called *occupancyMatrix* is created. This matrix stores count of the patient available at that time interval for each day and each room.

In order to start scheduling, we started from day 1 and found all of the patients that have the same day property. After patients are filtered, individuals which have highest priority among them are chosen. In this array of chosen patients, one with highest operation duration is selected. Our algorithm works in a way that, if the individual with longest duration is scheduled first, rest of the small ones could be arranged to fit between them and achieve the desired highest patient value.

Patient with highest priority and longest operation duration is shifted along its available interval until it finds a interval that minimizes the number of collisions with other available time intervals. If there is no interval to fit, in other words total number of collisions is equal to the infinity, next room is tested for same procedure. At the end if there is no room available operation is postponed unless it is already postponed. If it is scheduled, it sets the parameters of the operation class and sets the start of the operation as decided starting time. Before starting to schedule another patient, the values in the *occupancyMatrix* at that day, selected room and scheduled interval is set to the infinity and values in the rest of the matrix at that day and at unselected rooms in available interval is reduced by 1.

With that algorithm, each operation is scheduled at the interval where there is minimum available interval. With this heuristic total of 85 patient is scheduled and 75 of them are scheduled to the their initial available time interval.

Utilization of the rooms are not our first priority in objective function value, however in order to do as much operation as we can, we have also keep our utilization also high. The values for the utilization could be seen in Table.1.1.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | DAY 1 | DAY 2 | DAY 3 | DAY 4 | DAY 5 | ROOM AVG | | ROOM 1 | 0.8750 | 0.8333 | 10.000 | 0.8333 | 0.9167 | 0.8917 | | ROOM 2 | 0.8750 | 0.8333 | 0.7083 | 0.7083 | 0.9167 | 0.8083 | | ROOM 3 | 0.7500 | 0.8333 | 0.6667 | 0.5000 | 0.8333 | 0.7167 | | DAY AVG | 0.8333 | 0.8333 | 0.7917 | 0.6806 | 0.8889 | 0.8056 | |

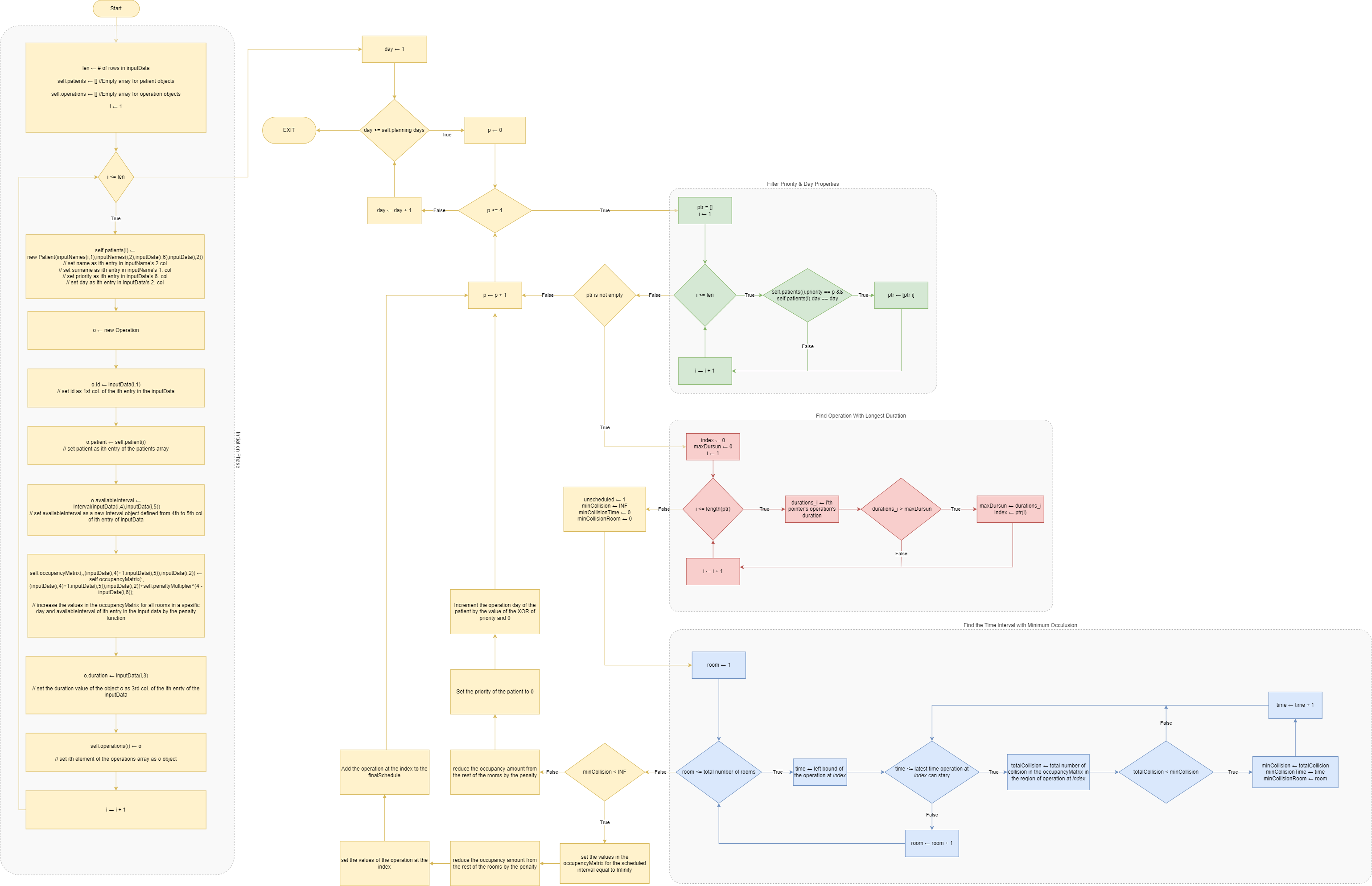
Table.1.1 Utilization of the rooms each day in heuristic 1

On the average, each room is utilized at least 70%. For each day average utilization is between 68% to 89%. In each day utilization is maximized for room number 1 since each scheduling procedure is starting from there. With the same manner, room three is utilized minimum for 5 days, since there is no patient left to fulfill this room.

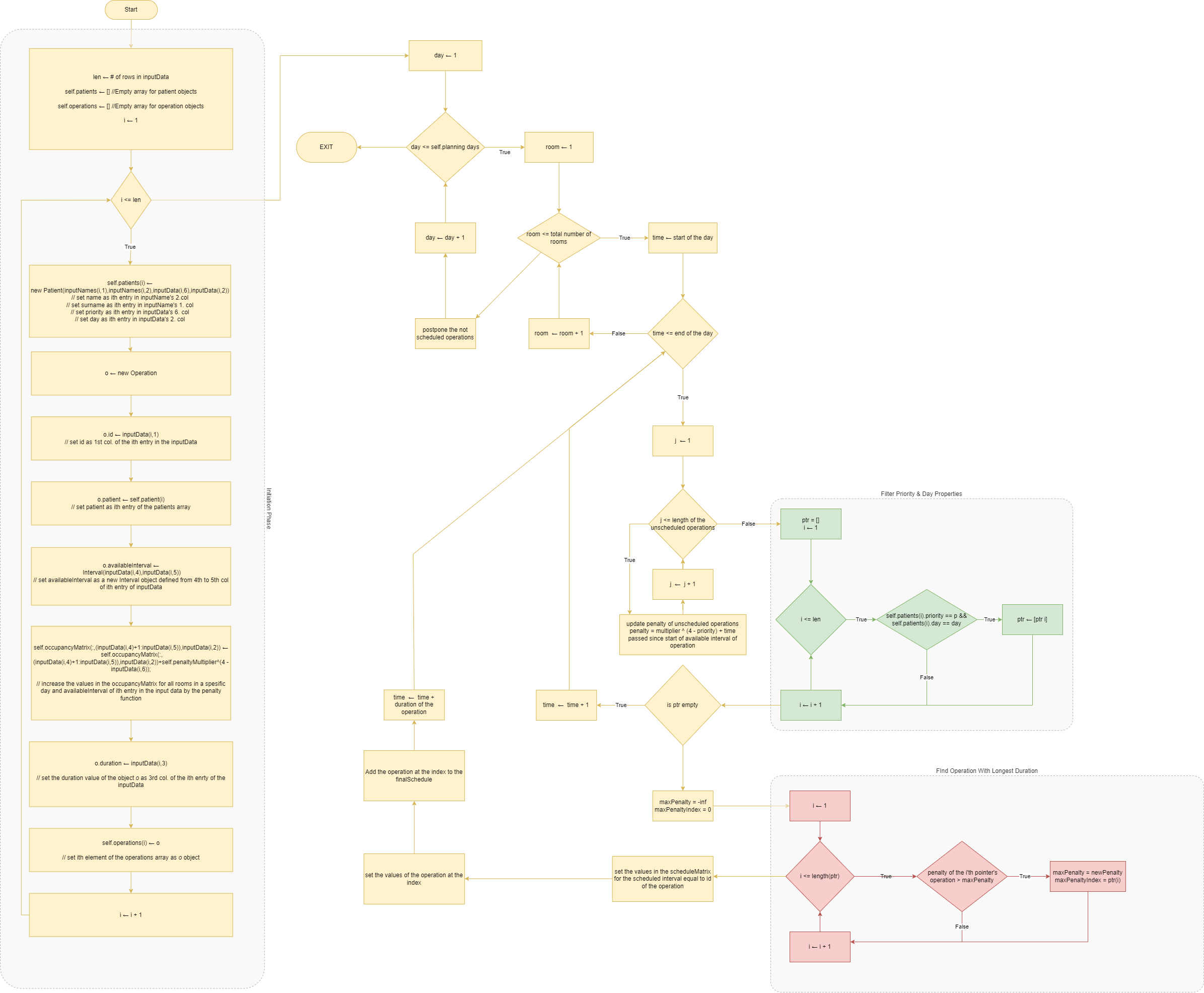
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **PRIORITY 1** | **PRIORITY 2** | **PRIORITY 3** | **PRIORITY 4** |
| **NUMBER OF OPERATIONS SHIFTED** | 20 | 18 | 10 | 6 |

Table.1.2 The number of operations shifted in each priority levels

For each priority number of operations shifted can be seen in Table.1.2. As can be seen on the table, as the priority gets higher more of the patients are shifted. Since patients with higher priority scheduled before rest of the other patients, they are fitted into the least occupied interval. This enables other low priority patients to fit in that spaces to maximize the number of the patients scheduled.

 Flowchart of the Algorithm

Objective 2

 Flowchart of the Algorithm

Conclusion

In this project, we designed algorithms based on heuristics for different objective functions and used these algorithms to schedule operations in a three-room local hospital. Based on approaches and scheduling obtained in this project, decision-makers of the local hospital could make their decision regarding the use of resources.

One of the findings of this project is that……

………..