

# Instruction for Computing Program I

## CS225 Group D2

Ren, Junjie; Chen, Yang; Cai, Peiqi; Li, Xinzhuo

### 1. Main description

#### 1.1 Assumptions

We assume that the registry center receives 10 patients for twice (that is, 20 patients a day in total), while every hospital can cure 5 people a day. There are in total 3 hospitals, that is, we can cure 15 patients in total.

Additionally, we randomly give the people in our input list risk status in the ratio of no: low: medium: high = 4:3:2:1. Basically, our input list (*register.csv*) provides 1500 random patients with their personal information.

Also, every patient has a unique id, that is the order of them in the input list, starting from 1.

#### 1.2 Priority Descriptions

Priority: 0. DDL (different); 1. Risk; 2. Prof; 3.age; 4. Date/time

For deadline, is register date + constant  $C$  ( $C$  is same for everyone). We assume the constant  $C$  to be 20.

In case of medium risk, remove from queue, and add to the queue after 20 days.

### 2. Registration

We assume every day first 10 patients insert into local queue 1 (morning), last 10 patients insert into local queue 2 (afternoon).

Every patient has a relative class *person*. It stores the patient's:

Id, name, address, phone, WeChat, email, profession, birth, risk

Timestamp, the date and the time of the registration

1 to 8 profession categories (I II III IV V VI VII VIII)

Seven age groups 12 18 35 50 65 75

Four risks 0 1 2 3

Location preferred (which hospital to choose)

Every time the centralized queue is ready to receive patients, registration queue checks: people who change risk, medium people who waits for 20 days, high risk people who wait for the centralized queue to be empty, re-registered people who have waited for 20 days, at last is the normal patients in the input list. (We assumed the priority.)

### 3. Queue

The centralized queue is built as a Fibonacci heap. And every node in the Fibonacci heap stores the pointer of the "person" class. In this case, we can save abundant space and the information of the patients are easy to renew or update. Some functions are added except those that usually exist in an ordinary Fibonacci heap. The queue receives the pointers of the patients by the function "record\_in" and

orders them by the function "comparePriority". Then it can provide a patient with the function "record\_out". Furthermore, there are some other functions that cooperate with the main function, such as removing all of the patients reaching DDL, changing the profession and risk of the patient, withdrawal, and so on. All in all, the names of the functions explain them well, otherwise, there should be some detailed comments.

#### **4. Appointment**

For every day, we first check whether there are some people approaching the deadline. If so, we treat it right away, no matter what their priority is. Under the circumstance that over 15 people have deadline today, we move the exceeding ones to the city town since the daily capacity in the local hospitals is fixed. On the other hand, if less than 15 people approach deadline, we pop the person with highest priority in the Fibonacci heap until local hospitals have been filled or the Fibonacci heap gets empty.

Then according to the preference, we choose the appropriate hospital and time slots for every person treated locally.

#### **5. Main program and Reporting**

##### **Weekly Report**

At the end of each week, we need a weekly report that is printed on the screen (Basically, output to a file is of the same function here, so we just use "cout" to print them out). What to print out is shown below:

The treated people

Registered people with appointment

Queueing people without appointments

(Including their name + ID + profession category + age category + risk + time)

##### **Monthly Report**

At the end of each month, we need a monthly report that is also printed on the screen. In monthly report, we will not print the people one by one. On the other side, we will count the total number of each group and print them out. What to print out is shown below:

Number of registered people this month

Number of waiting people (in the central queue now)

Number of appointments in all three hospitals of people who have withdrawn

##### **Main**

- i. We partly used the random generated data from Li Zihao (from GitHub, open source), we also modified it a lot, such as risk, profession generation, preferred hospital generation.
- ii. Registry: With two local queues daily containing all the registered people in between and then put them into the centralized queue.
- iii. Each day:

Push waiting people to the Fibonacci queue

Hospitals deal with the people.

Manage patients (update, withdraw, re-register): We provide some operations at the start of the day to remain the rights to update the professions and risks. Also, they can withdraw and register again after they have withdrawn but we will give them some two weeks delay unless they are in medium or high risk.

iv. Weekly report/monthly report

#### **Main Instruction**

When running the program, just follow the instruction of the terminal, read them carefully and choose the operation you want. We have tried our best to make it user-friendly.