1.4 Question 3

You are tasked with designing a record-keeping system for a hospital. A `Patient` recor their unique `patient_id` and the `urgency` of the condition they were admitted with. I there is an `intake_time` associated with each patient. The hospital would like you to data structures to optimize their access to the information for specific purposes. You w then "architect" the second one for a future engineer to implement. You are only allowe in the STL libraries already included in the `main.cpp`.

Implement

You will be implementing the following data structures. You should not need to impleme declaration of standard STL data structures (pick wisely) and the last loop to print out th

Part I) Ordered by intake times

The hospital frequently needs to query patients by their `intake_time`s or to look at a r bookkeeping purposes. They want a data structure that holds all of the patients organiz

- Quick access by intake-time
- Ordered by intake-time to allow statistics to be calculated on a range of input time

Part II) Patient access

Hospital staff frequently needs to query a patient by their unique `patient_id`. They do
ordered in any fashion, but they do need quick access to a patient by their unique `pati`

Architect

You will not be implementing this data structure. You will instead be providing commen
`main.cpp` file to specify what would be necessary to implement this portion. This shou
next engineer will have no problem getting started on the implementation.

Part III) "Ordered" by urgency

The hospital would also like a system to be built (eventually) which will loosely order the
means that a patient with a `High` urgency condition will be treated before all patients w
those patients will be treated before all patients with `Low` urgency conditions. You will n

- Annotate what STL data structure should be declared (and why)
- Annotate what operation should be used to add the patients to this data structure
  be passed in, and what the run-time of this operation would be.
- Annotate any additional functions that would need to be defined. You don't need r
  simply what the high-level functionality would be.

Annotations

You will need to complete the annotations in the comments of the `main.cpp` file for th
analysis of your code.

Output

At the very end, you will need to output the list of patients with each patient on a new lin
earliest intake-time (smallest) to most recent intake-time (largest).

For `smallInput.txt` you should get exactly the following result:

```
Sorted by intake times:
Glenn Murphy (307926022): High
Gladys Erickson (659679839): Medium
Byron Stevenson (754627995): Medium
Gary Horton (383692046): Low
Beatrice Nash (901650004): High
Marco Moran (675859534): Low
Tony Baker (13641386): High
```

```
Domingo Rogers (630805271): Low
Sergio Shaw (479718635): Low
Grady Martin (709873501): Low
Jean Summers (349554263): Medium
Tanya Torres (270151579): Low
Laurie Gomez (712303146): Low
Florence Bailey (642388602): Low
Jeanne Alvarez (874088477): Low
```

LAB ACTIVITY | 1.4.1: Question 3

Submission Instructions

Downloadable files

`main.cpp` , `Patient.cpp` , `Patient.h` , and `smallInput.txt`

Compile command

```
g++ main.cpp Patient.cpp -Wall -o a.out
```

We will use this com

Upload your files below by dragging and dropping into the area or choosing

main.cpp

Drag file here
or
Choose on hard drive.

Submit for grading

Latest submission - 11:13 PM on 03/18/20 — Submission passed all tests ✓

☐ Only show failing tests — **Downloa**

1: Patient ordering ^