DS2030 Data Structures and Algorithms for Data Science
Lab 5

September 16th, 2025

Lab Instructions

- Create a folder named **"DS2030_<RollNo.>"** (all letters in capital) in **"home"** directory. Eg- **DS2030_142402022**
- Name the script files in the given format **"<your_roll_no>_<Name>_Lab5.py"**
- Make sure the **folder, files, classes, functions and attributes** are named as instructed in the lab sheet.
- We will not be able to evaluate your work if the folder is not properly named or is not located in the home directory.
- Make sure to save your progress before leaving the lab.
- Do not shut down the system after completing the lab.
- You are not allowed to share code with your classmates nor allowed to use code from other sources.

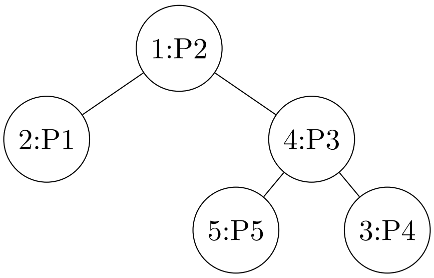
Patient Queue Management System for a hospital emergency department

Design a Patient Queue Management System for a hospital emergency department.

- Patients with critical conditions should be treated first. A lower priority number indicates a higher urgency (for example, priority = 1 is treated before priority = 3).
- The system should efficiently maintain the patients using an array-based heap, so that the most critical patient can be quickly identified and treated at any time.

Heap Visualization Example

Patients:



Attributes

Patient Heap stores:

- `patient_id` : Unique ID of the patient.
- `priority` : Severity level (integer, lower = more critical).

Methods

- add_patient(patient)
  - Add a new patient to the system
  - Restore the heap-order property
- get_next_patient()
  - Remove and return the most critical patient
  - Restore the heap-order property
- show_all_patients()
  - Display all patients currently in the system.
- update_priority(name, new_priority)
  - Find a patient by name and change their priority
  - Restore the heap-order property

Starter Code

```python
# Patient class representing each patient in the queue
class Patient:
    def __init__(self, priority, name):
        # Attributes:
        # priority (int) - lower value = more critical patient
        # name (str) - patient's name




# PatientQueue implements a min-heap based priority queue
class PatientQueue:
    def __init__(self):
        self.heap = []  # list of Patients

    # Add a new patient to the heap
    def add_patient(self, patient):
        """
        1. Append new patient to the end of the heap.
        2. Upheap to maintain min-heap property.
        """


    # Treat the patient with the highest priority
    def get_next_patient(self):
        """
        1. Take the root (patient with highest priority).
        2. Remove last patient and replace the root with last patient.
        3. Downheap to restore min-heap property.
        4. Return the treated patient.
        """


    # Update the priority of a specific patient
    def update_priority(self, name, new_priority):
        """
        1. Find the patient by name.
        2. Update their priority.
        3. Maintain the heap-order property by upheap/downheap as needed.
        """
    # Display all patients in heap order
    def show_all_patients(self):
        ”””
        Prints patients in current heap order with name and priority.
        ”””
```

Testing

```
def test_patient_queue():
    pq = PatientQueue()

    # Adding the new patients
    patients = [
        (4, ”Monica”), (5, ”Amit”), (6, ”Ivan”), (15, ”Mary”), (9, ”John”),
        (7, ”Abram”), (20, ”Vihan”), (16, ”Kevin”), (25, ”Nancy”), (14, ”Harshitha”),
        (12, ”Ravi”), (11, ”Varun”), (13, ”Karan”), (2, ”David”)
    ]

    for priority,name in patients:
        patient = Patient(priority, name)
        pq.add_patient(patient)

    print(”All Patients in heap order:”)
    pq.show_all_patients()
    print(”\nTreated Patient:”)
    treated = pq.get_next_patient()
    print(f”Name: {treated.name}, Priority: {treated.priority}\n”)

    print(”\nRemaining Patients in heap order:”)
    pq.show_all_patients()
    # Increase priority of Amit from 5 to 1 (higher priority)
    print(”\nUpdating priority of Amit from 5 to 1...”)

    pq.update_priority(”Amit”, 1)
    pq.show_all_patients()

    # Decrease priority of Abram to 17 (lower priority)
    print(”\nUpdating priority of Abram from 7 to 17...”)
    pq.update_priority(”Abram”, 17)
    pq.show_all_patients()


    print(”\nTreated Patient:”)
    treated = pq.get_next_patient()
    print(f”Name: {treated.name}, Priority: {treated.priority}”)
    print(”\nRemaining Patients in heap order:”)
    pq.show_all_patients()


# Run the updated test
test_patient_queue()
```

Sample Output

```
All Patients in heap order:
Name: David, Priority: 2
Name: Amit, Priority: 5
Name: Monica, Priority: 4
Name: Mary, Priority: 15
Name: John, Priority: 9
Name: Abram, Priority: 7
Name: Ivan, Priority: 6
Name: Kevin, Priority: 16
Name: Nancy, Priority: 25
Name: Harshitha, Priority: 14
Name: Ravi, Priority: 12
Name: Varun, Priority: 11
Name: Karan, Priority: 13
Name: Vihan, Priority: 20

Treated Patient:
Name: David, Priority: 2


Remaining Patients in heap order:
Name: Monica, Priority: 4
Name: Amit, Priority: 5
Name: Ivan, Priority: 6
Name: Mary, Priority: 15
Name: John, Priority: 9
Name: Abram, Priority: 7
Name: Vihan, Priority: 20
Name: Kevin, Priority: 16
Name: Nancy, Priority: 25
Name: Harshitha, Priority: 14
Name: Ravi, Priority: 12
Name: Varun, Priority: 11
Name: Karan, Priority: 13

Updating priority of Amit from 5 to 1...
Name: Amit, Priority: 1
Name: Monica, Priority: 4
Name: Ivan, Priority: 6
Name: Mary, Priority: 15
Name: John, Priority: 9
Name: Abram, Priority: 7
Name: Vihan, Priority: 20
Name: Kevin, Priority: 16
Name: Nancy, Priority: 25
Name: Harshitha, Priority: 14
Name: Ravi, Priority: 12
Name: Varun, Priority: 11
Name: Karan, Priority: 13

Updating priority of Abram from 7 to 17...
Name: Amit, Priority: 1
Name: Monica, Priority: 4
Name: Ivan, Priority: 6
Name: Mary, Priority: 15
Name: John, Priority: 9
Name: Varun, Priority: 11
Name: Vihan, Priority: 20
Name: Kevin, Priority: 16
Name: Nancy, Priority: 25
Name: Harshitha, Priority: 14
Name: Ravi, Priority: 12
Name: Abram, Priority: 17
Name: Karan, Priority: 13
```

```
Treated Patient:
Name: Amit, Priority: 1

Remaining Patients in heap order:
Name: Monica, Priority: 4
Name: John, Priority: 9
Name: Ivan, Priority: 6
Name: Mary, Priority: 15
Name: Ravi, Priority: 12
Name: Varun, Priority: 11
Name: Vihan, Priority: 20
Name: Kevin, Priority: 16
Name: Nancy, Priority: 25
Name: Harshitha, Priority: 14
Name: Karan, Priority: 13
Name: Abram, Priority: 17
```