DATA STRUCTURES AND ALGORITHMS

CS202M

PROJECT REPORT

INSTITUTE SEAT ALLOCATION SYSTEM FOR NEW ADMITS

Submitted by:

Shubhang S. Galagali

Ashish Manash

Submitted to:

Prof. Annappa B.

Professor,

Department of Computer Science and Engineering,

NITK Surathkal.

National Institute of Technology Karnataka [NITK], Surathkal.

NH 66, Srinivasnagar, Surathkal, Mangalore, Karnataka- 575025.

INSTITUTE SEAT ALLOCATION SYSTEM FOR NEW ADMITS

Team Members:

Name:	Shubhang S. Galagali	Ashish Manash
E-mail ID:	shubhanggalagali@gmail.com	ashishmanash1@gmail.com

Source Code : **main.py** @https://github.com/galava-shubhang/Institute_Seat_Allocation_System/

---

1. Problem Statement: (taken as was in the project proposal submitted)

'Develop a program which handles the task of seat allotment to students in an institute, according to the student's priority, among a list of participating institutes, accepting admissions through a common entrance examination.'

---

2. Introduction:

This Institute Seat Allocation System is designed to manage the allocation of seats for students based on their preferences, ranks, and available seats across institutes based on a common entrance examination criterion. This system is an attempt to simulate a real-world application of an admission process in educational institutions where students are ranked and seats are allocated based on the obtained ranks.

---

3. Primary Objectives:

The primary objectives of the project are:

- To develop a role-based access system for secure handling of users including Admin, Institute, and Student roles.
- To create a student and institute database, allowing for efficient data management and retrieval.
- To maintain a priority queue structure for effective sorting based on student ranks.
- To implement a seat allocation mechanism that respects student rank and preferences.
- To build a functional, interactive dashboard for each user role (Admin, Institute, and Student).

---

4. Design Overview: The project is structured into various components that support role management, data storage, rank-based sorting, preference handling, and allocation processes.

Each component is outlined below:

4.1 Defining Roles and Permissions

The project defines roles with specific permissions:

- **Admin**: Has full access to the system, including viewing all user data, institutes, and final seat allotments.

- **Institute**: Can view students admitted to the institute under different branches, along with student ranks.
- **Student**: Can register, login, and view their details, preferences, and allocated seats.

4.2 User and Student Database

- **User Database**:
  Uses a dictionary data structure to manage user credentials (username and password) for different roles.
- **Student Database**:
  Stores information on students' ranks and preferences in a python dictionary. Preferences are stored as a list of institute-branch pairs, and random rank assignments simulate a real entrance exam scenario.

Note: In python, Dictionaries are preferred for database-like operations for several key reasons:

- Key-Value pairs provide natural mapping for real-world data.
- Hash table implementation implies memory efficiency, no need of contiguous memory like arrays.
- Unique keys prevent duplicates
- Easy to maintain references, add/remove fields.
- O(1) time complexity in the average case.

4.3 Student Preferences and Rank Generation

The snippet generates:

- **Ranks**: Unique, randomized ranks for each student, simulating a real-world rank distribution scenario.
- **Preferences**: Realistic preferences for each student, with students allowed to select a minimum of 3 and a maximum of 5 choices among different institutes and branches.

Note: 4.3 was undertaken due to lack of a real database of students and their choices.

4.4 Priority Queue Implementation for Rank-Based Sorting

A priority queue organizes students based on ranks, ensuring that higher-ranked students are processed first during seat allocation. This implementation maintains the sorted order of students in ascending rank, with lower ranks receiving higher priority.

4.5 Seat Allocation System

The seat allocation system considers each student's preferences in order. For each student in the priority queue:

- The system checks the seat availability in the preferred institute and branch.
- If a seat is available, the student is allocated to that branch in that institute, and the seat count is decremented.

This approach prioritizes high-ranked students' choices and allocates seats based on their preferences as much as possible.

4.6 Dashboard Functions

The project implements three different dashboards, allowing users to interact with the system based on their role:

- **Admin Dashboard**: Displays all student data, institute details, and final seat allotments – all using Python dictionaries.
- **Institute Dashboard**: Shows allocated students for the respective institute, organized by branch and student rank, in a python dictionary.
- **Student Dashboard**: Displays the student's ID, rank, preferences, and allocated seat, if any.

---

5. Implementation:

The following outlines the project's core implementation aspects.

5.1 Role-Based Login System

Only users with valid credentials can access their respective dashboards. Implementing a login() function which checks if the username and password entered by the user matches the data stored/initialised in the system under 'users' dictionary as objects of the 'User' class.

5.2 Registration System for New Users

A registration function enables new student users to register in the system. Implementing a register_user() function which creates a new object in the system under 'users' dictionary as objects of the 'User' class, and initialises using the input given by the user.

Currently, this new registered user cannot participate in the allocation process, but in the next version of the project, we shall implement in such a manner where using database management tools, new users can be allowed to register and participate in the allocation process.

5.3 Priority Queue for Student Sorting

The priority queue is implemented as a list-based insertion sort, allowing rank-based organization. Each student ID and rank pair is enqueued, maintaining ascending order of ranks to ensure efficient allocation processing.

PriorityQueue Class:

- Initialization (_init_): Initializes an empty list (queue) to store student data.
- Check if Empty (is_empty): Returns True if the queue is empty, allowing easy checking of the queue's state.
- Enqueue Method (enqueue): Takes student_id and rank as parameters. Inserts a student into the queue while maintaining sorted order based on rank. This is achieved by iterating through the queue to find the correct position and then using list.insert() to add the student.
- Dequeue Method (dequeue): Removes and returns the student with the highest priority (lowest rank).
- If the queue is empty, it returns None; otherwise, it pops the first element from the list.

5.4 Seat Allocation Logic

Seats are defined for each participating institute and branch, with seat capacities initialized to 10 per branch. There are 5 participating institutes, all of them being engineering institutions, namely, IITB, IITM, NITC, NITK, and NITT. Each institute is assumed to have 2 branches, namely EC (Electronics and Communication Engineering) and ME (Mechanical Engineering. The allocation algorithm processes students in priority order, assigning seats to the highest-ranking students based on their top available preferences. The priority queue gives the ID of the student having the highest priority, i.e., numerically the lowest rank in the queue, that student's preferences are fetched from the students' database, and the algorithm for allocation runs until either the student is allocated a seat as per his ordered preference or exists if either preferences are exhausted or all seats of each preference is already filled.

5.5 Dashboard Functionality

Each role has specific access permissions:

- Admin Dashboard: Accesses the entire student database, participating institutes and final seat allotments. This is done by calling and printing the dictionary that holds the student details, and the final seat allotment results.
- Institute Dashboard: Views admitted students for specific institutes. This is achieved by creating a new dictionary from the seat allotments dictionary where only those students who have obtained the respective institute are filtered using the institute username and displayed.
- Student Dashboard: Allows students to view their ranks, preferences, and allocated seats. This is implemented by calling the specific items held in the dictionary under the key represented by the respective student usernames.

Note: **Student database** – It has been initialised for simplicity in a python dictionary as objects of class 'Student', but it can be taken as input from the user during registration and can be stored in a database, as in a .csv file. Currently, the username and the password of the student is initialised to their registration number, starting from 24101 upto 24199. Also, we have assumed 300 students took up the common entrance examination, so the minimum rank is 1, and maximum rank is 300 (numerically), with valid student IDs going from 24101 upto 24400. 'random' function has been used for generating the ranks and also the preferences in a shuffled manner.

---

4. Testing: PTO

All testcases were run locally on Jupyter Notebooks, on a Lenovo-IdeaPad-Laptop.

Testcases showcasing Registering a new student, ID outside of 24101 to 24199 initialised by default (above) and Logging into that newly registered account.

```
1. Admin Login (A)
2. Institute Login (I)
3. Student Register/Login (S)
4. Exit (E)

Choose an option:  S
Enter student username:  24200
Welcome Student! Create a new password for your account!
Enter student password:  24200
Please enter your rank:  255
Please enter your preferences!  [('NITK', 'EC'), ('NITT', 'ME')]
Registration successful!

Kindly re-login to access your account!
```

```
1. Admin Login (A)
2. Institute Login (I)
3. Student Register/Login (S)
4. Exit (E)

Choose an option:  S
Enter student username:  24200
Enter student password:  24200

Login Successful!


 Student dashboard

Student ID: 24200,
 Student Rank: 255,

 Student Preferences: [('NITK', 'EC'), ('NITT', 'ME')],
 Allocated Seat: None
```

Testcase showing logging in using the Admin role – All students data, participating institutes and the final seat allotments are displayed.

```
1. Admin Login (A)
2. Institute Login (I)
3. Student Register/Login (S)
4. Exit (E)

Choose an option:  A
Enter admin username:  Admin
Enter admin password:  DSAPROJECTCS202M

Login successful!

 Admin dashboard

 Student Database - ADMIN access:
{24101: 24101,24101,247,[('NITT', 'ME'), ('NITT', 'EC'), ('IITB', 'EC'), ('NITC', 'EC')],User is a student, 24102: 24102,24102,156,[('NITK',
'ME'), ('IITB', 'ME'), ('NITC', 'ME'), ('IITB', 'ME'), ('NITC', 'EC')],User is a student, 24103: 24103,24103,210,[('NITT', 'ME'), ('IITM', 'E
C'), ('IITM', 'EC')],User is a student, 24104: 24104,24104,261,[('NITT', 'EC'), ('IITB', 'ME'), ('NITK', 'ME'), ('NITK', 'ME')],User is a stu
dent, 24105: 24105,24105,33,[('IITB', 'EC'), ('IITB', 'ME'), ('IITB', 'ME'), ('NITT', 'ME'), ('NITC', 'ME')],User is a student, 24106: 24106,
24106,116,[('NITT', 'ME'), ('IITB', 'ME'), ('IITB', 'EC'), ('IITM', 'ME'), ('NITC', 'ME')],User is a student, 24107: 24107,24107,126,[('NIT
K', 'ME'), ('NITC', 'EC'), ('IITB', 'ME')],User is a student, 24108: 24108,24108,245,[('NITT', 'ME'), ('NITC', 'ME'), ('NITT', 'ME'), ('IIT
B', 'EC'), ('IITB', 'ME')],User is a student, 24109: 24109,24109,115,[('NITC', 'ME'), ('NITK', 'EC'), ('NITK', 'ME'), ('NITK', 'ME'), ('NIT
C', 'ME')],User is a student, 24110: 24110,24110,208,[('IITM', 'EC'), ('IITB', 'ME'), ('IITB', 'EC')],User is a student, 24111: 24111,24111,1
44,[('IITB', 'EC'), ('NITT', 'EC'), ('IITB', 'ME'), ('IITB', 'EC')],User is a student, 24112: 24112,24112,139,[('NITK', 'ME'), ('NITC', 'M
E'), ('IITM', 'ME')],User is a student, 24113: 24113,24113,39,[('NITC', 'EC'), ('NITC', 'EC'), ('NITC', 'EC'), ('IITB', 'EC')],User is a stud
ent, 24114: 24114,24114,185,[('IITM', 'EC'), ('NITC', 'EC'), ('IITM', 'EC'), ('NITT', 'EC')],User is a student, 24115: 24115,24115,51,[('NIT
T', 'ME'), ('NITT', 'ME'), ('IITB', 'EC'), ('IITB', 'ME'), ('NITT', 'ME')],User is a student, 24116: 24116,24116,266,[('IITM', 'ME'), ('IIT
M', 'EC'), ('IITM', 'ME'), ('NITC', 'ME')],User is a student, 24117: 24117,24117,66,[('NITT', 'ME'), ('IITB', 'ME'), ('NITK', 'ME'), ('NITK',
'EC')],User is a student, 24118: 24118,24118,6,[('IITB', 'ME'), ('NITK', 'ME'), ('IITM', 'EC')],User is a student, 24119: 24119,24119,63,[('N
ITC', 'ME'), ('NITK', 'EC'), ('IITM', 'ME'), ('NITK', 'EC')],User is a student, 24120: 24120,24120,69,[('NITC', 'ME'), ('IITB', 'ME'), ('IIT
M', 'EC'), ('NITC', 'EC')],User is a student, 24121: 24121,24121,288,[('IITB', 'ME'), ('IITB', 'ME'), ('NITK', 'ME'), ('NITT', 'ME')],User is
a student, 24122: 24122,24122,56,[('NITK', 'EC'), ('IITM', 'EC'), ('IITM', 'EC'), ('NITT', 'EC'), ('NITK', 'ME')],User is a student, 24123: 2
4123,24123,61,[('NITK', 'EC'), ('IITM', 'ME'), ('NITT', 'EC')],User is a student, 24124: 24124,24124,42,[('IITB', 'ME'), ('NITC', 'EC'), ('NI
TC', 'EC')],User is a student, 24125: 24125,24125,206,[('NITT', 'ME'), ('NITT', 'EC'), ('NITC', 'EC')],User is a student, 24126: 24126,24126,
71,[('NITK', 'EC'), ('IITB', 'ME'), ('NITC', 'ME')],User is a student, 24127: 24127,24127,88,[('NITC', 'ME'), ('IITB', 'ME'), ('NITK', 'M
E')],User is a student, 24128: 24128,24128,35,[('NITC', 'EC'), ('IITB', 'EC'), ('NITC', 'EC'), ('IITM', 'ME'), ('NITT', 'ME')],User is a stud
ent, 24129: 24129,24129,147,[('NITK', 'EC'), ('IITB', 'EC'), ('IITB', 'ME')],User is a student, 24130: 24130,24130,215,[('NITK', 'EC'), ('NIT
T', 'ME'), ('NITK', 'EC'), ('NITK', 'EC'), ('NITT', 'EC')],User is a student, 24131: 24131,24131,208,[('IITB', 'ME'), ('IITB', 'EC'), ('IIT
M', 'EC'), ('IITM', 'EC'), ('NITC', 'ME')],User is a student, 24132: 24132,24132,135,[('NITT', 'ME'), ('NITT', 'ME'), ('IITM', 'ME'), ('NIT
T', 'ME')],User is a student, 24133: 24133,24133,234,[('NITC', 'EC'), ('NITC', 'EC'), ('NITK', 'ME')],User is a student, 24134: 24134,24134,1
48,[('NITK', 'EC'), ('NITK', 'ME'), ('IITB', 'ME')],User is a student, 24135: 24135,24135,49,[('NITT', 'ME'), ('IITM', 'ME'), ('IITB', 'ME'),
('IITM', 'ME')],User is a student, 24136: 24136,24136,248,[('NITT', 'EC'), ('IITM', 'ME'), ('NITC', 'EC'), ('IITM', 'EC'), ('IITM', 'ME')],Us
er is a student, 24137: 24137,24137,111,[('NITC', 'ME'), ('IITM', 'ME'), ('NITC', 'ME')],User is a student, 24138: 24138,24138,10,[('NITC',
'EC'), ('NITC', 'ME'), ('IITM', 'EC')],User is a student, 24139: 24139,24139,154,[('NITT', 'ME'), ('IITM', 'ME'), ('NITC', 'EC')],User is a s
tudent, 24140: 24140,24140,184,[('IITB', 'ME'), ('NITC', 'EC'), ('NITK', 'ME')],User is a student, 24141: 24141,24141,65,[('IITM', 'EC'), ('N
ITC', 'ME'), ('IITM', 'ME'), ('NITT', 'EC')],User is a student, 24142: 24142,24142,79,[('NITC', 'ME'), ('IITB', 'ME'), ('IITB', 'EC'), ('NIT
C', 'ME')],User is a student, 24143: 24143,24143,43,[('IITM', 'EC'), ('IITM', 'ME'), ('IITB', 'EC')],User is a student, 24144: 24144,24144,2
3,[('IITB', 'ME'), ('NITT', 'EC'), ('NITT', 'EC'), ('NITK', 'EC'), ('NITC', 'EC')],User is a student, 24145: 24145,24145,214,[('IITB', 'ME'),
('NITK', 'EC'), ('IITM', 'ME')],User is a student, 24146: 24146,24146,13,[('IITB', 'EC'), ('NITC', 'ME'), ('NITC', 'ME'), ('NITT', 'ME')],Use
r is a student, 24147: 24147,24147,119,[('IITM', 'ME'), ('NITT', 'EC'), ('NITK', 'ME')],User is a student, 24148: 24148,24148,192,[('NITK',
'ME'), ('NITT', 'ME'), ('NITK', 'ME')],User is a student, 24149: 24149,24149,5,[('IITM', 'EC'), ('NITK', 'ME'), ('NITK', 'ME'), ('IITB', 'E
C'), ('NITT', 'EC')],User is a student, 24150: 24150,24150,182,[('NITT', 'EC'), ('NITK', 'ME'), ('IITB', 'EC'), ('NITT', 'EC')],User is a stu
dent, 24151: 24151,24151,288,[('NITC', 'ME'), ('NITT', 'EC'), ('NITT', 'ME')],User is a student, 24152: 24152,24152,132,[('IITM', 'ME'), ('NI
TK', 'EC'), ('NITC', 'EC')],User is a student, 24153: 24153,24153,124,[('NITK', 'EC'), ('IITM', 'EC'), ('NITK', 'EC'), ('IITB', 'EC')],User i
s a student, 24154: 24154,24154,123,[('IITB', 'EC'), ('IITM', 'EC'), ('IITM', 'EC'), ('NITT', 'ME'), ('IITM', 'EC')],User is a student, 2415
5: 24155,24155,146,[('NITT', 'ME'), ('IITM', 'ME'), ('IITB', 'EC'), ('NITC', 'EC'), ('NITK', 'ME')],User is a student, 24156: 24156,24156,31,
[('IITM', 'ME'), ('NITK', 'ME'), ('NITK', 'ME')],User is a student, 24157: 24157,24157,159,[('NITC', 'ME'), ('NITC', 'EC'), ('IITM', 'EC'),
('NITK', 'ME'), ('NITC', 'EC')],User is a student, 24158: 24158,24158,296,[('IITB', 'ME'), ('NITK', 'EC'), ('NITT', 'ME')],User is a student,
24159: 24159,24159,227,[('IITM', 'EC'), ('IITB', 'EC'), ('NITC', 'EC')],User is a student, 24160: 24160,24160,9,[('NITC', 'M
E'), ('NITC', 'ME'), ('NITK', 'ME'), ('NITT', 'ME'), ('NITC', 'ME')],User is a student, 24161: 24161,24161,90,[('NITT', 'EC'), ('NITK', 'E
C'), ('NITK', 'EC'), ('NITK', 'ME')],User is a student, 24162: 24162,24162,145,[('NITT', 'ME'), ('NITC', 'ME'), ('IITB', 'EC'), ('IITB', 'M
E')],User is a student, 24163: 24163,24163,282,[('NITT', 'ME'), ('NITC', 'EC'), ('IITM', 'EC'), ('IITM', 'ME'), ('NITK', 'EC')],User is a stu
dent, 24164: 24164,24164,24,[('IITM', 'ME'), ('NITK', 'EC'), ('NITT', 'ME')],User is a student, 24165: 24165,24165,84,[('NITT', 'EC'), ('NIT
K', 'ME'), ('IITB', 'ME')],User is a student, 24166: 24166,24166,175,[('NITT', 'ME'), ('IITB', 'EC'), ('NITT', 'ME'), ('IITM', 'EC')],User is
a student, 24167: 24167,24167,299,[('NITK', 'EC'), ('NITK', 'ME'), ('NITT', 'EC'), ('NITC', 'ME')],User is a student, 24168: 24168,24168,278,
[('NITC', 'EC'), ('NITK', 'ME'), ('IITM', 'ME'), ('NITC', 'EC')],User is a student, 24169: 24169,24169,226,[('NITT', 'EC'), ('NITT', 'ME'),
('NITT', 'EC'), ('NITT', 'ME'), ('NITC', 'EC')],User is a student, 24170: 24170,24170,53,[('NITK', 'EC'), ('IITB', 'EC'), ('NITC', 'ME'), ('I
ITB', 'ME'), ('NITT', 'ME')],User is a student, 24171: 24171,24171,194,[('NITC', 'EC'), ('NITK', 'EC'), ('IITB', 'ME')],User is a student, 24
172: 24172,24172,82,[('NITT', 'ME'), ('NITK', 'EC'), ('IITB', 'EC')],User is a student, 24173: 24173,24173,18,[('IITM', 'ME'), ('IITB', 'M
E'), ('IITM', 'EC')],User is a student, 24174: 24174,24174,219,[('NITT', 'ME'), ('NITC', 'ME'), ('NITT', 'EC'), ('IITM', 'ME'), ('NITC', 'M
E')],User is a student, 24175: 24175,24175,180,[('NITT', 'EC'), ('NITT', 'EC'), ('NITT', 'EC'), ('IITM', 'ME')],User is a student, 24176: 241
76,24176,141,[('IITB', 'ME'), ('IITB', 'EC'), ('IITM', 'EC'), ('IITM', 'ME'), ('NITK', 'ME')],User is a student, 24177: 24177,24177,107,[('II
TM', 'ME'), ('NITC', 'ME'), ('NITK', 'ME'), ('IITB', 'ME'), ('NITK', 'ME')],User is a student, 24178: 24178,24178,191,[('IITB', 'EC'), ('NIT
T', 'ME'), ('IITB', 'ME')],User is a student, 24179: 24179,24179,190,[('IITM', 'EC'), ('NITT', 'EC'), ('NITT', 'ME'), ('NITT', 'ME'), ('NIT
T', 'EC')],User is a student, 24180: 24180,24180,32,[('NITK', 'ME'), ('NITC', 'EC'), ('NITT', 'ME'), ('IITM', 'EC')],User is a student, 2418
1: 24181,24181,58,[('NITC', 'EC'), ('IITB', 'EC'), ('IITB', 'ME'), ('NITT', 'EC')],User is a student, 24182: 24182,24182,284,[('NITK', 'EC'),
('IITB', 'EC'), ('NITT', 'ME'), ('IITB', 'ME'), ('IITM', 'EC')],User is a student, 24183: 24183,24183,137,[('NITC', 'EC'), ('NITK', 'ME'),
('NITC', 'ME'), ('IITM', 'ME'), ('IITM', 'EC')],User is a student, 24184: 24184,24184,78,[('IITB', 'ME'), ('NITC', 'ME'), ('IITM', 'EC'), ('N
ITC', 'ME')],User is a student, 24185: 24185,24185,170,[('NITT', 'ME'), ('IITM', 'ME'), ('NITK', 'ME')],User is a student, 24186: 24186,2418
6,20,[('IITM', 'EC'), ('NITT', 'EC'), ('NITT', 'ME')],User is a student, 24187: 24187,24187,295,[('NITC', 'ME'), ('NITT', 'E
C'), ('IITB', 'ME')],User is a student, 24188: 24188,24188,300,[('IITB', 'EC'), ('IITM', 'ME'), ('NITT', 'ME'), ('IITB', 'ME')],User is a stu
dent, 24189: 24189,24189,22,[('NITT', 'ME'), ('IITB', 'EC'), ('NITT', 'ME'), ('IITB', 'ME')],User is a student, 24190: 24190,24190,101,[('IIT
B', 'EC'), ('NITC', 'ME'), ('IITM', 'EC')],User is a student, 24191: 24191,24191,134,[('NITC', 'ME'), ('IITM', 'ME'), ('NITT', 'EC'), ('IIT
B', 'ME'), ('NITK', 'EC')],User is a student, 24192: 24192,24192,212,[('IITB', 'ME'), ('IITM', 'EC'), ('IITB', 'EC'), ('NITC', 'EC'), ('IIT
B', 'ME')],User is a student, 24193: 24193,24193,179,[('NITK', 'ME'), ('NITT', 'EC'), ('NITK', 'ME'), ('NITK', 'ME'), ('IITM', 'EC')],User is
a student, 24194: 24194,24194,151,[('NITK', 'ME'), ('IITM', 'EC'), ('NITC', 'EC')],User is a student, 24195: 24195,24195,112,[('IITB', 'ME'),
('NITC', 'EC'), ('IITM', 'EC')],User is a student, 24196: 24196,24196,70,[('NITC', 'EC'), ('NITC', 'ME'), ('IITM', 'EC'), ('NITT', 'ME'), ('N
ITC', 'ME')],User is a student, 24197: 24197,24197,185,[('NITT', 'EC'), ('NITC', 'ME'), ('NITK', 'EC'), ('IITB', 'ME')],User is a student, 24
198: 24198,24198,252,[('NITK', 'EC'), ('IITM', 'EC'), ('NITT', 'ME'), ('IITM', 'ME'), ('IITB', 'ME')],User is a student, 24199: 24199,24199,4
7,[('NITT', 'EC'), ('NITC', 'ME'), ('NITT', 'ME')],User is a student, 24200: 24200,24200,255,[('NITK', 'EC'), ('NITT', 'ME')],User is a stude
nt}

 List of participating institutes:
 IITB,
 IITM,
 NITC,
 NITK,
 NITT.

 Final Seat Allotments of each student:
{24101: ('NITT', 'EC'), 24102: ('NITK', 'ME'), 24103: ('IITM', 'EC'), 24104: ('NITK', 'ME'), 24105: ('IITB', 'EC'), 24106: ('NITT', 'ME'), 24
107: ('NITK', 'ME'), 24108: ('IITB', 'EC'), 24109: ('NITC', 'ME'), 24110: ('IITM', 'EC'), 24111: ('IITB', 'EC'), 24112: ('NITK', 'ME'), 2411
3: ('NITC', 'EC'), 24114: ('IITM', 'EC'), 24115: ('NITT', 'ME'), 24116: ('IITM', 'ME'), 24117: ('NITT', 'ME'), 24118: ('IITB', 'ME'), 24119:
('NITC', 'ME'), 24120: ('NITC', 'ME'), 24121: ('IITB', 'ME'), 24122: ('NITK', 'EC'), 24123: ('NITK', 'EC'), 24124: ('IITB', 'ME'), 24125: ('N
ITT', 'EC'), 24126: ('NITK', 'EC'), 24127: ('NITC', 'ME'), 24128: ('NITC', 'EC'), 24129: ('NITK', 'EC'), 24130: ('NITK', 'EC'), 24131: ('IIT
B', 'ME'), 24132: ('NITT', 'ME'), 24133: ('NITC', 'EC'), 24134: ('NITK', 'EC'), 24135: ('NITT', 'ME'), 24136: ('NITT', 'EC'), 24137: ('NITC',
'ME'), 24138: ('NITC', 'EC'), 24139: ('NITT', 'ME'), 24140: ('IITB', 'ME'), 24141: ('IITM', 'EC'), 24142: ('NITC', 'ME'), 24143: ('IITM', 'E
C'), 24144: ('IITB', 'ME'), 24145: ('NITK', 'EC'), 24146: ('IITB', 'EC'), 24147: ('IITM', 'ME'), 24148: ('NITK', 'ME'), 24149: ('IITM', 'E
C'), 24150: ('NITT', 'EC'), 24151: None, 24152: ('IITM', 'ME'), 24153: ('NITK', 'EC'), 24154: ('IITB', 'EC'), 24155: ('NITT', 'ME'), 24156:
('IITM', 'ME'), 24157: ('NITC', 'ME'), 24158: None, 24159: ('IITM', 'EC'), 24160: ('NITC', 'ME'), 24161: ('NITT', 'EC'), 24162: ('NITT', 'M
E'), 24163: ('NITC', 'EC'), 24164: ('IITM', 'ME'), 24165: ('NITT', 'EC'), 24166: ('IITB', 'EC'), 24167: ('NITK', 'ME'), 24168: ('NITC', 'E
C'), 24169: ('NITT', 'EC'), 24170: ('NITK', 'EC'), 24171: ('NITC', 'EC'), 24172: ('NITT', 'ME'), 24173: ('IITM', 'ME'), 24174: ('NITC', 'M
E'), 24175: ('NITT', 'EC'), 24176: ('IITB', 'ME'), 24177: ('IITM', 'ME'), 24178: ('IITB', 'EC'), 24179: ('IITM', 'EC'), 24180: ('NITC', 'M
E'), 24181: ('NITC', 'EC'), 24182: ('NITK', 'EC'), 24183: ('NITC', 'EC'), 24184: ('IITB', 'ME'), 24185: ('IITM', 'ME'), 24186: ('IITM', 'E
C'), 24187: None, 24188: ('IITB', 'EC'), 24189: ('NITT', 'ME'), 24190: ('IITB', 'EC'), 24191: ('NITC', 'ME'), 24192: ('IITB', 'ME'), 24193:
('NITK', 'ME'), 24194: ('NITK', 'ME'), 24195: ('IITB', 'ME'), 24196: ('NITC', 'EC'), 24197: ('NITT', 'EC'), 24198: ('IITM', 'EC'), 24199: ('N
ITT', 'EC')}
```

Testcases showing New Student Registration – Invalid Rank and Exit Option.

```
1. Admin Login (A)
2. Institute Login (I)
3. Student Register/Login (S)
4. Exit (E)

Choose an option:  S
Enter student username:  24230
Welcome Student! Create a new password for your account!
Enter student password:  24230
Please enter your rank:  500
Invalid rank. Rank should be between 1 and 300.

Kindly re-login to access your account!
```

```
1. Admin Login (A)
2. Institute Login (I)
3. Student Register/Login (S)
4. Exit (E)

Choose an option:  E
Exiting...
```

Testcases showing Logging in as a student and accessing allotted seat (above) and Invalid Password (below).

```
1. Admin Login (A)
2. Institute Login (I)
3. Student Register/Login (S)
4. Exit (E)

Choose an option:  S
Enter student username:  24116
Enter student password:  24116

Login Successful!


 Student dashboard

Student ID: 24116,
 Student Rank: 266,

 Student Preferences: [('IITM', 'ME'), ('IITM', 'EC'), ('IITM', 'ME'), ('NITC', 'ME')],
 Allocated Seat: ('IITM', 'ME')
```

```
1. Admin Login (A)
2. Institute Login (I)
3. Student Register/Login (S)
4. Exit (E)

Choose an option:  S
Enter student username:  24168
Enter student password:  24160
Invalid username or password!! Please try again.
```

Testcases showing Login of Student and accessing his allotted seat (upper portion) and Invalid Password for Institute Login (below).

```
1. Admin Login (A)
2. Institute Login (I)
3. Student Register/Login (S)
4. Exit (E)

Choose an option:  S
Enter student username:  24101
Enter student password:  24101

Login Successful!


 Student dashboard

Student ID: 24101,
 Student Rank: 247,

 Student Preferences: [('NITT', 'ME'), ('NITT', 'EC'), ('IITB', 'EC'), ('NITC', 'EC')],
 Allocated Seat: ('NITT', 'EC')
```

```
1. Admin Login (A)
2. Institute Login (I)
3. Student Register/Login (S)
4. Exit (E)

Choose an option:  I
Enter institute username:  NITK
Enter institute password:  NITT
Invalid username or password!! Please try again.
```

```
1. Admin Login (A)
2. Institute Login (I)
3. Student Register/Login (S)
4. Exit (E)

Choose an option:  I
Enter institute username:  NITK
Enter institute password:  NITK
Login successful!

 Institute dashboard

{'ME': [(24102, 156), (24104, 261), (24107, 126), (24112, 139), (24148, 192), (24167, 299), (24180, 32), (24193, 179), (24194, 151)], 'EC': [(24122,
56), (24123, 61), (24126, 71), (24129, 147), (24130, 215), (24134, 148), (24145, 214), (24153, 124), (24170, 53), (24182, 284)]}
```

Testcase showing successful Institute Login and Institute Dashboard (above).

Testcases showing invalid Role choice, successful institute login (IITM) and invalid Admin Password, successful institute login (IITB) (in respective order below).

```
1. Admin Login (A)
2. Institute Login (I)
3. Student Register/Login (S)
4. Exit (E)

Choose an option:  IITM
Invalid choice.
```

```
1. Admin Login (A)
2. Institute Login (I)
3. Student Register/Login (S)
4. Exit (E)

Choose an option:  I
Enter institute username:  IITM
Enter institute password:  IITM
Login successful!

 Institute dashboard

{'EC': [(24103, 210), (24110, 298), (24114, 185), (24141, 65), (24143, 43), (24149, 5), (24159, 227), (24179, 190), (24186, 20), (24198, 292)], 'M
E': [(24116, 266), (24147, 119), (24152, 132), (24156, 31), (24164, 24), (24173, 18), (24177, 107), (24185, 170)]}
```

```
1. Admin Login (A)
2. Institute Login (I)
3. Student Register/Login (S)
4. Exit (E)

Choose an option:  A
Enter admin username:  admin
Enter admin password:  dsapro
Invalid username or password!! Please try again.
```

```
1. Admin Login (A)
2. Institute Login (I)
3. Student Register/Login (S)
4. Exit (E)

Choose an option:  I
Enter institute username:  IITB
Enter institute password:  IITB
Login successful!

 Institute dashboard

{'EC': [(24105, 33), (24108, 246), (24111, 144), (24146, 13), (24154, 123), (24166, 175), (24178, 191), (24188, 300), (24190, 101)], 'ME': [(24118,
6), (24121, 208), (24124, 42), (24131, 200), (24140, 184), (24144, 23), (24176, 141), (24184, 78), (24192, 212), (24195, 112)]}
```

SOURCE CODE FOR THE PROJECT

GitHub Link: https://github.com/galava-shubhang/Institute_Seat_Allocation_System/

END OF REPORT