DS2030 Data Structures and Algorithms for Data Science
Lab 4

September 9th, 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>_Lab4.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.

Phone Directory using Binary Search Tree (BST)

You are required to implement a **Phone Directory** using a **Binary Search Tree (BST)**. Each entry in the directory corresponds to a student and will be stored in a BST node.

Attributes

Each node should store the following:

- `name` : Name of the student.
- `number` : Phone number.
- `roll_number` : Roll number of the student (string, used as the key for BST).
- `email` : Email ID.
- `left` : Left child.
- `right` : Right child.

Methods

- `add_contact(name, number, roll_number, email)` : Insert a new contact.
- `remove_contact(roll_number)` : Delete a contact by name.
- `search(root, roll_number)` : Search for a contact and return details if found.
- `show_directory()` : Display all contacts in alphabetical order.
- `update_contact(roll_number, name=None, number=None, email=None)` : **Challenge** — Update details of a contact if it exists.
- `node_depth(roll_number)`: find the depth of the node in the tree.
- `node_height(roll_number)`: find the height of the node in the tree.

Starter Code

```
class ContactNode:
    def __init__(self, name, number, roll_number, email):
        """
        Input:
            name (str): Name of the contact
            number (str): Phone number of the contact
            roll_number (str): Unique roll number of the contact
            email (str): Email address of the contact
        Output:
            Initializes a ContactNode object with left and right set to None
        """
        self.name = name
        self.number = number
        self.roll_number = roll_number
        self.email = email
        self.left = None
        self.right = None


class PhoneDirectory:
    def __init__(self):
        """
        Input:  None
        Output:   Initializes the PhoneDirectory with root set to None
        """
        self.root = None

    def insert(self, root, node):
        """
        Input:
            root (ContactNode): Current root node of the BST
            node (ContactNode): Node to be inserted
        Output:
            Returns the root of the BST after insertion
        """
        pass

    def add_contact(self, name, number, roll_number, email):
        """
        Input:
            name (str): Name of the contact
            number (str): Phone number of the contact
            roll_number (str): Unique roll number of the contact
            email (str): Email address of the contact
        Output:
            Inserts a new contact into the directory; returns nothing
        """
        pass

    def search(self, root, roll_number):
        """
        Input:
            root (ContactNode): Current root node of the BST
            roll_number (str): Roll number of the contact to search for
        Output:
            Returns the ContactNode if found, else None
        """
        pass

    def delete(self, root, roll_number):
        """
        Input:
            root (ContactNode): Current root node of the BST
            roll_number (str): Roll number of the contact to delete
        Output:
            Returns the root of the BST after deletion
        """
        pass

    def remove_contact(self, roll_number):
        """
        Input:  roll_number (str): Roll number of the contact to delete
        Output:   Deletes the contact from the directory; returns nothing
        """
        pass

    def inorder(self, root):
        """
        Input:  root (ContactNode): Current root node of the BST
        Output: Prints the contacts in ascending order of roll_number
        """
        pass

    def show_directory(self):
        """
        Input:  None
        Output: Prints all contacts in sorted order by roll number
        """
        pass

    def update_contact(self, roll_number, name=None, number=None, email=None):
        """
        Input:
            roll_number (str): Roll number of the contact to update
            name (str, optional): New name to update
            number (str, optional): New phone number to update
            email (str, optional): New email address to update
        Output:
            Updates the contact details if found; returns True if updated, else False
        """
        pass

    def node_depth(self, roll_number):
        """
        Input:  roll_number (str): Roll number of the node to find the depth
        Output: Returns the depth of the node if found, else -1
        """
        pass

    def node_height(self, roll_number):
        """
        Input:  roll_number (str): Roll number of the node to find the height
        Output: Returns the height of the node if found, else -1
        """
        pass
```

Testing

```python
def test():
    directory = PhoneDirectory()
    directory.add_contact("Bob", "8882220002", "142401005", "142401005@smail.iitpkd.ac.in")
    directory.add_contact("Alice", "9991110001", "142401001", "142401001@smail.iitpkd.ac.in")
    directory.add_contact("Charlie", "7773330003", "142401008", "142401008@smail.iitpkd.ac.in")

    print("Phone Directory (Inorder):")
    directory.show_directory()

    print("\nSearching for 142401005:")
    contact = directory.search(directory.root, "142401005")
    if contact:
        print("Found:", contact.name, contact.number, contact.roll_number, contact.email)
    else:
        print("Not Found")

    print("\nUpdating Charlie's phone number...")
    directory.update_contact("142401008", number="7000000000")
    directory.show_directory()

    print("\nDeleting Alice...")
    directory.remove_contact("142401001")
    print("Updated Directory (Inorder):")
    directory.show_directory()

    print("\nNode Depth for Bob:")
    depth = directory.node_depth("142401005")
    print("Depth:", depth)

    print("\nNode Height for Charlie:")
    height = directory.node_height("142401008")
    print("Height:", height)
test()
```

Sample Output

```
Phone Directory (Inorder):
Alice   9991110001 142401001 142401001@smail.iitpkd.ac.in
Bob     8882220002 142401005 142401005@smail.iitpkd.ac.in
Charlie 7773330003 142401008 142401008@smail.iitpkd.ac.in

Searching for 142401005:
Found: Bob 8882220002 142401005 142401005@smail.iitpkd.ac.in

Updating Charlie's phone number...
Alice   9991110001 142401001 142401001@smail.iitpkd.ac.in
Bob     8882220002 142401005 142401005@smail.iitpkd.ac.in
Charlie 7000000000 142401008 142401008@smail.iitpkd.ac.in

Deleting Alice...
Updated Directory (Inorder):
Bob     8882220002 142401005 142401005@smail.iitpkd.ac.in
Charlie 7000000000 142401008 142401008@smail.iitpkd.ac.in

Node Depth for Bob:
Depth: 1

Node Height for Charlie:
Height: 0
```