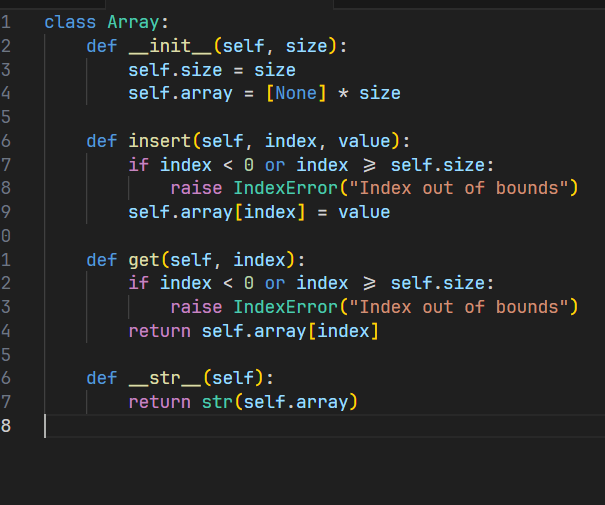
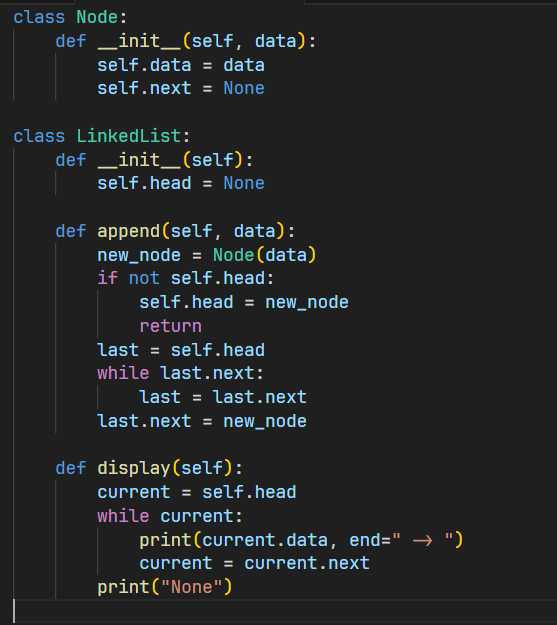
**I.Data structure:**

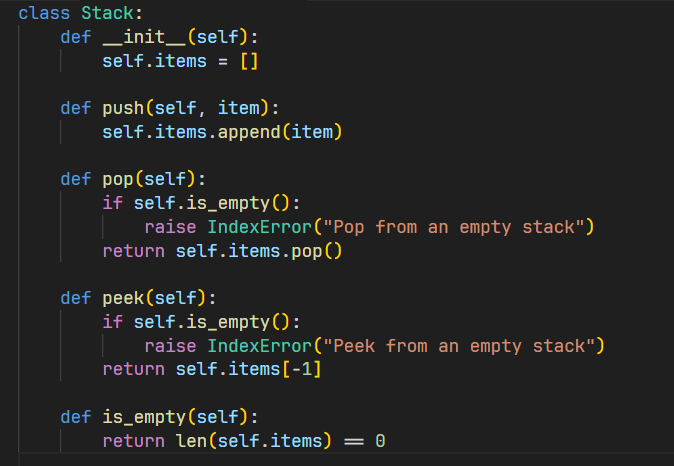
**1.Array**



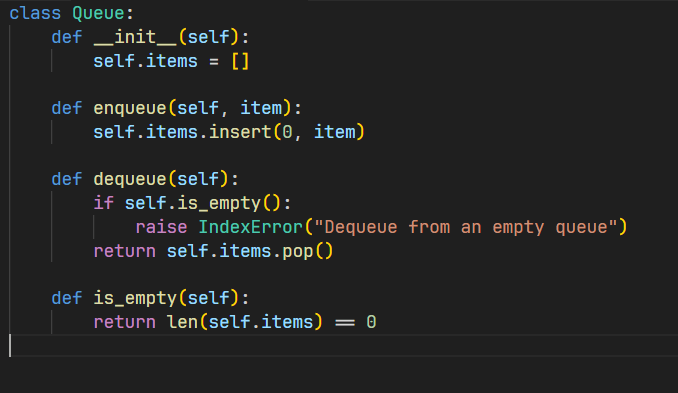
**2. Linked list :**

****

**3. Stack :**

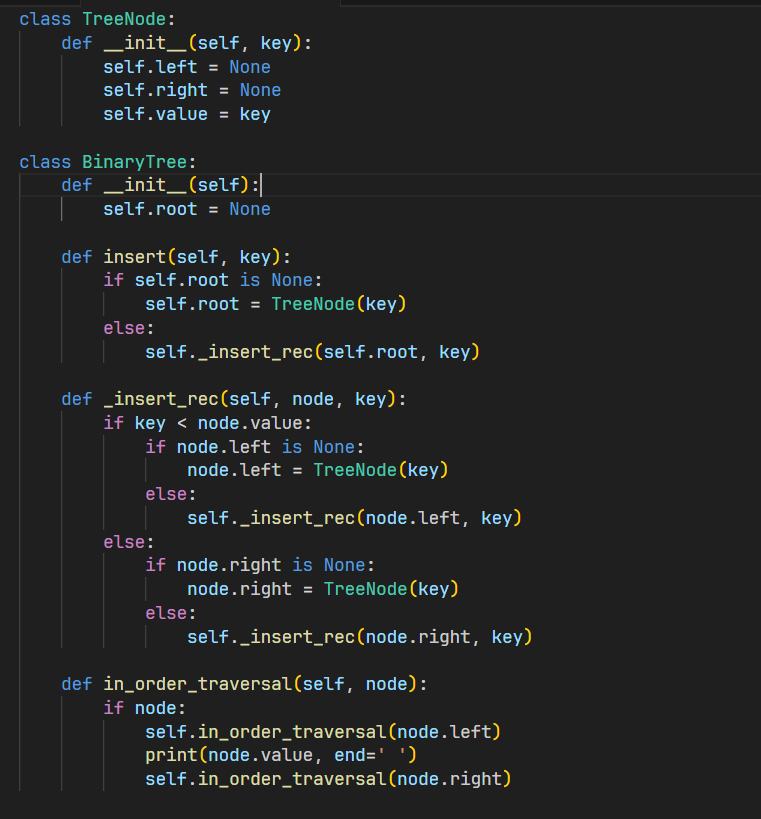
****

**4. Queue:**

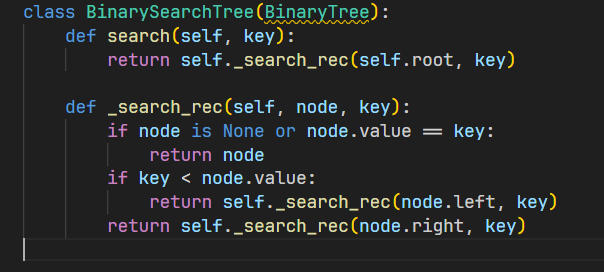
****

**II.** **Tree Implementations**

**1.Binary tree:**

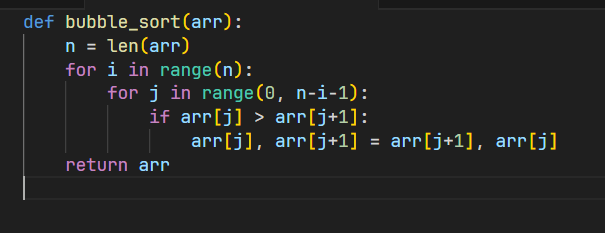
****

**2. Binary Search Tree :**

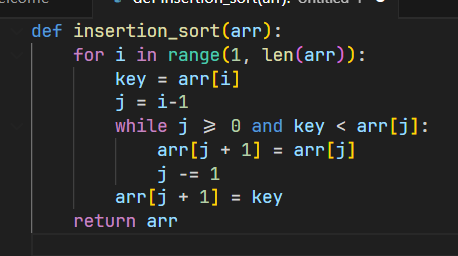
****

**III. Sorting Algorithms :**

1. **Bubble sort :**

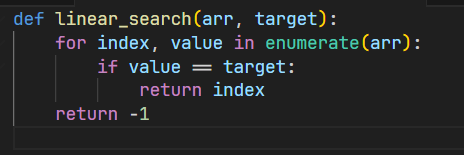
****

1. **Insertion sort :**

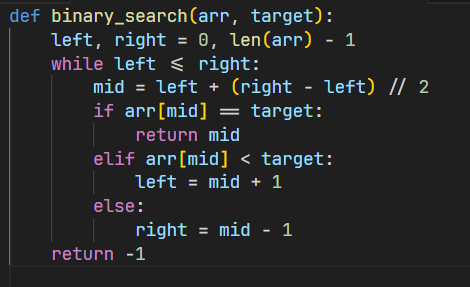
****

**IV. Searching Algorithms :**

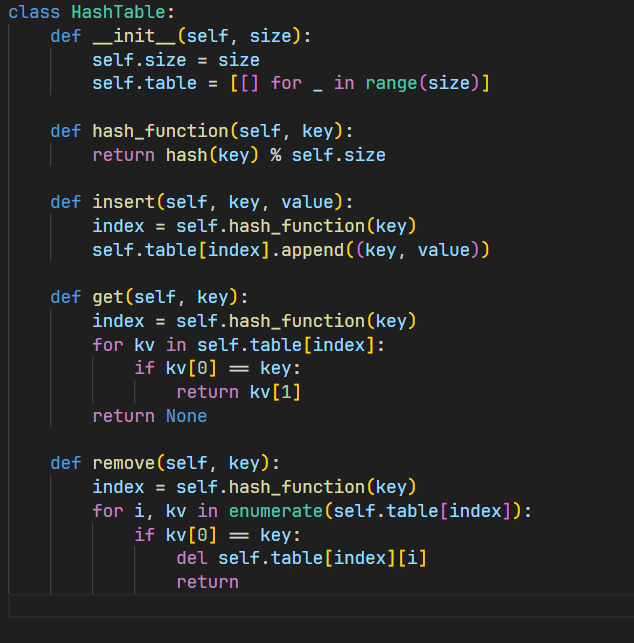
1. **Linear Search :**

****

1. **Binary search :**

****

**V. Hash table implementation :**

****

| **Data Structure / Algorithm** | **Time Complexity (Best)** | **Time Complexity (Average)** | **Time Complexity (Worst)** | **Space Complexity** |
| --- | --- | --- | --- | --- |
| **Array (Access)** | **O(1)** | **O(1)** | **O(1)** | **O(n)** |
| **Linked List (Insertion)** | **O(1)** | **O(n)** | **O(n)** | **O(n)** |
| **Stack (Push/Pop)** | **O(1)** | **O(1)** | **O(1)** | **O(n)** |
| **Queue (Enqueue/Dequeue)** | **O(1)** | **O(1)** | **O(1)** | **O(n)** |
| **Binary Tree (Insertion)** | **O(log n)** | **O(n)** | **O(n)** | **O(n)** |
| **Binary Search Tree (Search)** | **O(1)** | **O(log n)** | **O(n)** | **O(n)** |
| **Bubble Sort** | **O(n)** | **O(n^2)** | **O(n^2)** | **O(1)** |
| **Insertion Sort** | **O(n)** | **O(n^2)** | **O(n^2)** | **O(1)** |
| **Linear Search** | **O(1)** | **O(n)** | **O(n)** | **O(1)** |