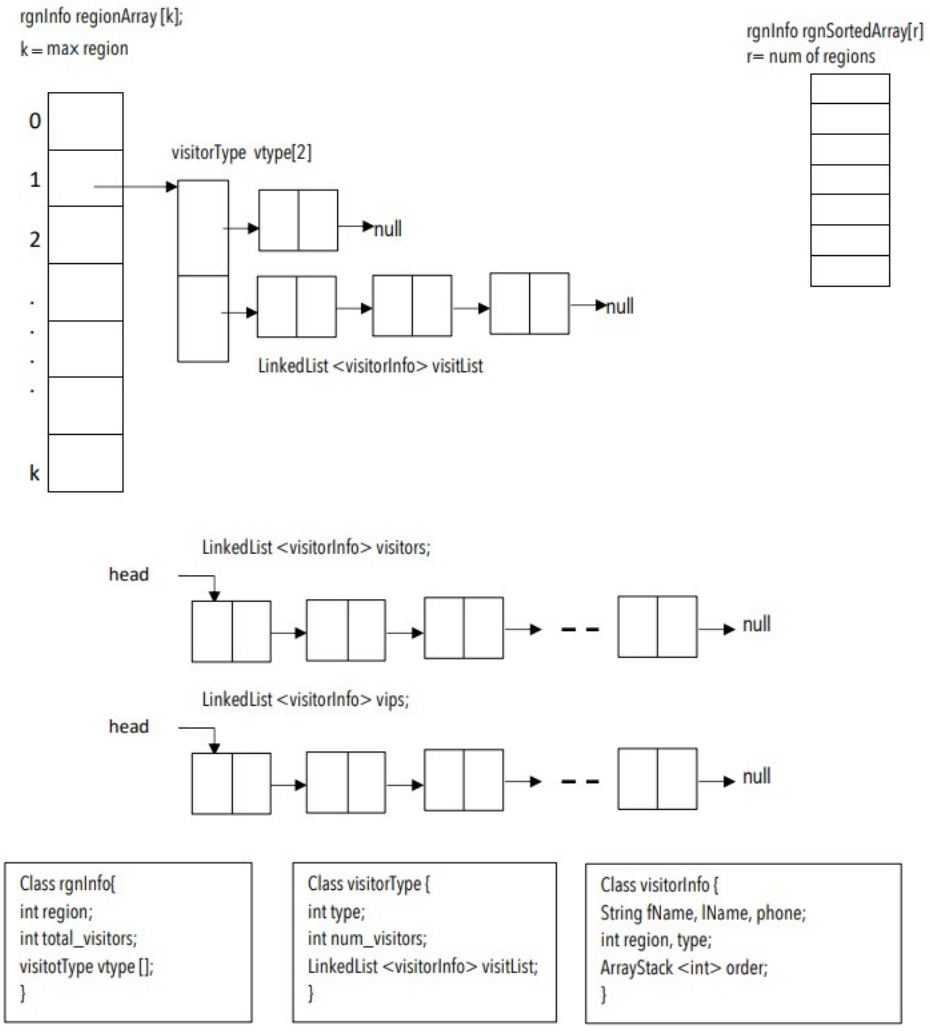
Data Structures (CSC212)
Second Trimester 2022/2023
Course Project
Phase-1

a)



b)

- **visitors** is a list that will be used to store all visitors' information in the order they were read from the file.

- **vips** is a list that will be used to store all VIP visitors' information in the order they were read from the file.

- **regionArray** is an array that will be used to store visitors' information based on their regions such that each index *i* will contain a list of VIP and regular pass holders coming from region *i*.

- **vtype** is an array of *visitorType* of size 2, where each index contains a list of type *visitorInfo* about each visitor type. Index 0 contains the information of the VIP pass holders while index 1 contains the information of the regular pass holders.

- **visitList** is a list of type *visitorInfo* where each node contains the visitor information such as the name, phone number, region, type of pass, and a stack including the order of visited kingdoms.

- **rgnSortedArray** is an array of type *rgnInfo* containing the regions ordered from highest to lowest according to their total number of visitors.

c)

Specification **ThemeParkADT**
Elements: **rgnInfo, visitorType, visitorInfo, Node, LinkedList**
Structure: See the graphical representation.
Operations:

0. Method ***readFileAndAnalyse(String f)***
Input: f
Requires: f should be an existing file with a valid filename.
Results: This method will open file f, read the text in the file line by line and analyze the text as follows: create a new node containing the visitor's information and add it to a list that contains all visitors. If it's a VIP visitor, it's also added to the vips list. Then, the array of regions is searched for the region r. If r is found, access it and insert the created node in the array of visitorType either in index 0 or 1 depending on the visitor's pass type. If r is not found, it's created at index r, containing the region's information and then the created node is inserted in the same way described previously. In the end, an array of type rgnInfo will be created with the regions sorted by the number of their visitors in a descending order.
Output: None

1. Method ***searchVisitor(String lName)***
Input: lName
Requires: None
Results: The method will print all visitors' information having last name *lName*.
Output: None

2. Method ***rgnCount()***
Input: None
Requires: None
Results: The method will print the total number of regions that the visitors come from.
Output: None

3. Method ***popularRgn()***
Input: None
Requires: None
Results: The method will print the regions and their number of visitors sorted by the total number of visitors in a descending order.
Output: None

4. Method ***vipRgn(int i)***
Input: int i
Requires: None
Results: The method will print the total number of VIP pass holders from region i.
Output: None

5. Method ***vipLocation()***
Input: None
Requires: None
Results: The method will print the current location of all VIP pass holders.
Output: None

6. Method ***checkVipLoc(String n1, String n2, Boolean flag)***
Input: n1, n2
Requires: visitors with phone numbers n1 and n2 are from the same region
Results: The method will return true if the two VIP visitors with phone numbers n1 and n2 are in the same Kingdom. Otherwise, a message will be printed indicating the reason the method is returning false.
Output: flag.

7. Method ***checkRegLoc(int r, String n1, String n2, Boolean flag)***
Input: r, n1, n2
Requires: None
Results: The method will return true if the two regular visitors with phone numbers n1 and n2 coming from region r visited the kingdoms in the same order. Otherwise, a message will be printed indicating the reason the method is returning false.
Output: flag.

d)

Method 1) *searchVisitor:* **O(n)**
Method 2) *rgnCount:* **O(1)**
Method 3) *popularRgn:* **O(r)**
Method 4) *vipRgn:* **O(1)**
Method 5) *vipLocation:* **O(v)**
Method 6) *checkVipLoc:* **O(v)**
Method 7) *checkRegLoc:* **O(g)**

Phase 2: (15 Marks) Due Date: (4 February 2023 11:59 pm)
In the second phase of the project, you will implement the provided ADT and the operations specified above using Java. Your program should contain a Main class that reads a specified text file and perform each of the seven operations that are described in this assignment. You will be asked to run a demo of your program and generate output using several test cases provided by the instructor. Failure to show for your demo will result in receiving ZERO for this phase.

Important notes:

• All data structures used in this assignment must be implemented by the students and any use of Java collections or any other libraries is strictly forbidden (you may only use the ADTs you have studied in the course).

• The program should implement the provided ADT. Implementing other representations will not be accepted.

• Submit your source code (compressed in ZIP format) through LMS. In addition, make sure to bring your LAPTOP, so you can present a demo of your work to your instructor