

Data Structure 2019

Lab 11

Mohammad Sadegh Najafi - Jonghyun Lee

May 24, 2019

Today's Task 1

Your first task is to implement **UGraph** and **DGraph** classes.

- **class UGraph** : Implement an unweighted, undirected graph using the Adjacency List representation
- **class DGraph**: Implement an unweighted, directed graph also using the Adjacency List representation
- You may change the type of `adjacencyList` if you wish.
 - Suggested type: `Map<Integer, List<Integer>> adjacencyList;`
- You should not change the **interface Graph** methods!
 - Except for the return type of `getAdjacencyList()`, `getVertices()`, and `getAdjVertices()` if you decide to change the type of `adjacencyList`
- You may add more classes, methods and data members if you feel it is necessary

Today's Task 2

Your second task is to implement BFS and DFS traversals.

- Implement the following methods
 - `public static Set<Integer> BFTraversal(Graph graph, int root)`
 - `public static Set<Integer> DFTraversal (Graph graph, int root)`
- Note:
 - They must return a **Set<Integer>** in the order of the traversals.
 - But remember that only LinkedHashSet keeps the order of added elements.
 - You may use any built-in Java data structures.
 - You may add helper methods if you feel it is necessary

Today's Task 3

Your third task is to complete the main function in the parts marked as

```
// Add code to construct the corresponding Graph ...
```

- **input.txt** contains the input for **UGraph**
 - The first line is the the number of vertices, starting from 1.
 - Starting from the second line, the first column in the file represents the vertex, and the particular row (other entries except the first column) tells all the vertices that the vertex is adjacent to
 - The vertices are separated by a **tab**!
- **input2.txt** contains the input for **DGraph**
 - The first line is the the number of vertices, starting from 1.
 - Starting from the second line, every row indicates an edge. The vertex label in first column is the tail and the vertex label in second column is the head. (e.g. "2 5" means 2 -> 5)
 - The vertices are separated by a **space**!

Expected Output (Lab11.java)

Vertex 0 - []

Vertex 1 - []

Vertex 2 - [3]

Vertex 3 - [1]

Vertex 4 - [0, 1]

Vertex 5 - [2, 0]

DFS traversal from 5 : [5, 0, 2, 3, 1] //or [5, 2, 3, 1, 0]

Vertex 0 - [5, 4]

Vertex 1 - [4, 3]

Vertex 2 - [5, 3]

Vertex 3 - [2, 1]

Vertex 4 - [0, 1]

Vertex 5 - [2, 0]

BFS traversal from 2 : [2, 5, 3, 0, 1, 4] //not the only possible answer

BFS

[1, 37, 79, 164, 155, 32, 87, 39, 113, ...

DFS

[1, 2, 5, 7, 10, ...

Submission

Submit your **Lab11.java** file by 5:40pm today on ETL.

Note that since your BFS and DFS traversals may produce different outcomes, we will use a separate method to check the correctness of your results with inputs such as **input.txt** and **input2.txt**. (We won't provide the checker methods)