

Homework 8

Djuro Radusinovic

171044095

Question 2

Problem-solution approach

I thought of 'two dimensional' linked list as if it was a grid. All the elements are connected inside in four directions up, down, prev(left), next(right) as it was noted in the homework. This is achieved through a node of this linked list that enables this by providing pointing references to each other. Of course it is generic and since question mentioned 'sorted' vertices this generic type will implement Comparable interface. Inside there are two lists of vertices, one points to the column vertices and the other points to row vertices. There are two addition classes and those are Vertex and Edge. Both of them extend the Node that is in list in order to be able to achieve interconnection. Graph has this list as a data field and uses it to store its elements. It can be either directed or undirected. It provides an iterator to iterate through the edges of a certain vertex as it is wanted. Vertex can be added or removed and when a vertex is removed, all of the edges containing it are removed. Edge can be added or removed. Breath and depth first searches are performed. I made the basic version of them and not the book's 'refined' one which returns the array of parents since it is easier to show the basic version of these searches in console by printing them. Of course, here I am printing the final order of the graph in string format.

Tests cases

Test cases are run inside the virtual machine provided. Its actual results can be confirmed from the attached screenshots.

Test Scenario	Expected Results	Actual Results
Creating a new graph and adding vertices A D E to it	New graph should be created	As expected
Adding vertices to this graph : AA, AD, AE, DA, DD, EE, DE, EA, ED	All these edges should be added	As expected
Using iterator to iterate through this graph's edges with source 'A'	Should be able to iterate using iterator's next and hasNext methods	As expected
Using breath first search traversal is performed in the following order:	Should traverse in following order : A D E	As expected

Using depth first search traversal is performed in the following order:	Should traverse in following order : E D A	As expected
Removing edge AD	Should remove it	As expected
Using breath first search traversal is performed in the following order:	After previous removal it should traverse in order: A E D	As expected
Removing edges EE, DE, AA, EE, AA in this order	EE, DE, AA should be removed and afterwards it should not alter the graph since they are repeating edges which are not now present in the graph	As expected
Removing vertex 'A'	Should remove it and its vertices	As expected
Trying to remove this vertex again	Should not be able to do it since it is not present	As expected
Constructing graph from homework's pdf	Should be constructed	As expected
Performing BFS for it	Traversal order should be A B E C D	As expected
Performing DFS for it	Traversal order should be C D E B A	As expected
Constructing one more graph. This time undirected one	Should be constructed	As expected
Inserting edge 5,2 and 7,4 to it	They should be added	As expected
Checking if edge 7,4 exists	Should exist since it is not directed	As expected
Removing an edge 2,5	Should be removed since it exists	As expected
Removing vertex 7	Should be removed	As expected
Checking if it is directed	Should not be directed	As expected

Class diagram


```
Seç Komut İstemi
B BA BE
C
D DA
E EA EC ED

Performing BFS for it(final order printed) - ABED should be printed
A B E C D
Performing DFS for it(final order printed) - CDEBA should be printed
C D E B A
Constructing a undirectional graph
Inserting edges 5,2 and 4,7
2 25
4 47
5 52
7 74

Checking if edge 7,4 exists
7,4 is edge
Removing an edge (2,5)
2 2
4 47
5 52
7 74

Removing vertex 7
2 2 4 5
4
5

Checks if it is directed
It is not directed

C:\Users\cse222\Desktop\hwk8\src>
```