**Lab 4.**

Examine the file graph.txt. It stores a graph that represents a social network. The syntax is that there is an edge form the node at position 1 to the node at position 3. You can ignore the values at position 2 and 4. Use the PageRank algorithm to rank the nodes. Use value of 0.9 for *d*. The algorithm should terminate when the L1 norm of the difference between the old and new numbers is less than 0.001. Print the top 20 most important nodes in ranked order starting with the most important.

Expected output:

[1159, 1293, 155, 55, 1051, 641, 729, 1153, 855, 323, 1245, 1260, 798, 1112, 1461, 963, 1463, 1306, 1179, 535]

**Design Considerations**

Store the graph as a set of nodes (HashSet<Integer>), an adjacency list (HashMap<Integer, ArrayList<Integer>>), and a variable that keeps track of the number out outgoing links for each node (HashMap<Integer, Integer>). The adjacency lists should store the set of **incoming** nodes for each node (i.e. the set of nodes that have edges to each node). Feel free to create a separate class for the graph that stores the graph data and graph methods. Keep track of two variables: the old and new values for the PageRank vector.

HashMap<Integer, Double> pageRankOld = new HashMap<>();  
HashMap<Integer, Double> pageRankNew = new HashMap<>();

Create a method that finds the L1 Norm between the two values (input may change depending on your implementation).

double findDistance(HashMap<Integer, Double> pageRankOld, HashMap<Integer, Double> pageRankNew)

Feel free to create new methods/classes for processing the input or performing an iteration of the algorithm. At the end of each iteration, use the clone method to make a copy of pageRankNew and assign it to pageRankOld. My solution is a total of about 100 lines of code.

If x=[x1,...xn] and y=[y1,...,yn] are vectors, the L1 norm of their distance is equal to |x1-y1|+...+|xn-yn|.

Note: When inserting and edge in the graph, please check for a preexisting edge between the two nodes. If an edge between the two nodes already exists, please do not add a new one.