Algorithm Design and Analysis

Assignment Three

Due **Friday 6 December** 2019

The purpose of this assignment is to augment a data structure (Question 1), and to create a useful utility that can analyze the configuration of a network for problems (Questions 2 and 3).

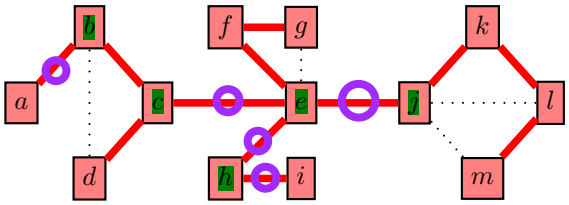
Questions:

1. Design and implement a data structure for holding integer values which has $O(\log_2 n)$ `add`, `contains` and `remove` methods, as well as an efficient `minGap` method that returns the difference between the two closest numbers currently in the data structure. Include a suitable `main` method to test the data structure. Hint: you can augment the class `RedBlackTree` that was prepared in the lab (an implementation of the `remove` method is not required). **(20 marks)**

2. Write a class called `GraphUtil` which includes a method:

   ```
   public GraphADT<Info<E>> calculateDepths
      (GraphADT<E> graph, Vertex<E> startVertex)
   ```

   which prepares the depth first tree that results from a depth first search starting at the specified start vertex, but whose `Info<E>` vertices contain a reference to the corresponding vertex v in the graph as well as integers $d(v)$ and $m(v)$. During the recursive depth first search of the graph a counter $d(v)$ is assigned to each vertex when it is first visited. The values $m(v)$ are recursively calculated during the depth first search as the minimum of the $d(w)$ values if there is an edge from v or a descendant of v to an already visited vertex w, or else it is taken to be $d(v)$, whichever is the smaller.

   For example, if in the illustrated diagram a depth-first search starting at a visits the vertices in the following order:

v	a	b	c	d	e	f	g	h	i	j	k	l	m
$d(v)$	0	1	2	3	4	5	6	7	8	9	10	11	12

then the following values of $m(v)$ are calculated:

v	a	b	c	d	e	f	g	h	i	j	k	l	m
$m(v)$	0	1	1	1	4	4	4	7	8	9	9	9	9

(15 marks)

3. Suppose G is a connected and undirected graph. An *articulation point* is a vertex whose removal would disconnect G, and a *bridge* is an edge whose removal would disconnect G. Articulation points and bridges are important in analyzing network reliability in the case of failure of a single router or network cable. Note that a non-root vertex v of the depth first tree is an articulation point if and only if it has a child w for which $m(w) \geq d(v)$. An edge from v to w that is included in the depth first tree is a bridge if and only if $m(w) > d(v)$.

   Enhance the class `GraphUtil` so that contains a method `analyzeGraph` which first checks that the graph is connected and undirected (and throws an exception if it is not). The method then analyzes the graph (treating the chosen start vertex of the depth first tree as a special case) and calculates all the articulation points and bridges in the graph. Include a suitable `main` method that clearly demonstrates the `analyzeGraph` method.

   (15 marks)