Individual tasks for homework 6

Goodrich, Tamassia - Data Structures and Algorithms in Java, 4th ed., Chapter 13

1. C-13.1
2. C-13.6
3. C-13.8
4. C-13.9
5. C-13.11
6. C-13.12
7. C-13.13
8. C-13.19
9. C-13.20
10. C-13.21
11. C-13.23
12. C-13.27
13. P-13.10

14. Find eccentricy of a given vertex
15. Find the diameter of a graph
16. Find the radius of a graph
17. Find the center of a graph
18. Find all bridges in a graph (removing a bridge makes graph disconnected)
19. Find the farthest vertex from a given vertex
20. Find the shortest path between two given vertices
21. Find the maximum flow between two given vertices
22. Find two vertices that have the longest distance between them
23. Find the minimum spanning tree of a graph

Requirements:

1. The program must be based on a given template (classes GraphTask, Graph, Vertex, Arc, ...). The result of the program must be a machine-processable object (list of arcs, list of vertices, graph with modified infofields, matrix, etc.) that can be used further, not just a printout. For example, a path in the graph must be a list of arcs (List<Arc>, not String).

2. If the task requires an undirected graph, then each edge AB is represented by two opposite arcs AB and BA.

3. The program must be tested with at least five relevant examples, including an example of a graph with 2000+ vertices (in which case only the execution time is measured).

4. It is advisable to print out a small meaningful example in the run method, where it would be possible to see what was given and what was the result without seeing the program text. The main content of the task should be formatted as a separate method, which receives the pre-generated initial data as a parameter and returns the result as a machine-processable object.

For example, the run method could be (let's say you need to find a path in a graph):

```
Graph g = .... // generate the source graph and other necessary data

List<Arc> path = g.mainMethod(parameters); // mainMethod solves your
task, but does not print. For big graphs measure the execution time

System.out.println ("Given graph: " + g + " with parameters " +
parameters);  // print the initial data

System.out.println ("Result: " + path); // print the result
```

All reports undergo automatic plagiarism check.

If a topic has already been chosen (check it on the forum, if you notice a recurring topic, let me know immediately), you need to find something else. The principle "First come, first served" applies and the later registrant must choose a new topic.

If you use the help of AI to create a report (this is allowed only for consultation, but not for creating text), then indicate which AI you asked which question to.