**Home exercise 5**

**Course name: Object oriented programming and design for engineering**

**Course number: 157109**

**Subject: Generics and Java Streams**

# Introduction: Cluster analysis and agglomerative clustering

From Wikipedia: ***Cluster analysis****or****clustering****is the task of grouping a set of objects in such a way that objects in the same group (called a****cluster****) are more similar (in some sense) to each other than to those in other groups (clusters). It is a main task of exploratory*[*data mining*](https://en.wikipedia.org/wiki/Data_mining)*, and a common technique for*[*statistical*](https://en.wikipedia.org/wiki/Statistics)[*data analysis*](https://en.wikipedia.org/wiki/Data_analysis)*, used in many fields, including*[*machine learning*](https://en.wikipedia.org/wiki/Machine_learning)*,*[*pattern recognition*](https://en.wikipedia.org/wiki/Pattern_recognition)*,*[*image analysis*](https://en.wikipedia.org/wiki/Image_analysis)*,*[*information retrieval*](https://en.wikipedia.org/wiki/Information_retrieval)*,*[*bioinformatics*](https://en.wikipedia.org/wiki/Bioinformatics)*,*[*data compression*](https://en.wikipedia.org/wiki/Data_compression)*, and*[*computer graphics*](https://en.wikipedia.org/wiki/Computer_graphics)*.*

There are many types of clustering algorithms, and you will implement an agglomerative clustering algorithm. The algorithm receives a set of elements that have a distance measure between them and a distance threshold that determines when two clusters are distinct. The algorithm pseudocode is given below:

Cluster(*items*, *threshold*)

*clusters* = A set of singletons, where each item starts in its own cluster

while (*clusters* is not of size 1)

find the two closest clusters, *c1*,*c2* from *clusters*

if (distance(*c1*, *c2*) > *threshold*) return *clusters*;

replace *c1*,*c2* in *clusters* with the union of c1,c2

return *clusters*

The distance between two clusters c1, c2 is defined as

i.e., it is the distance between the two closest items, the first from c1, and the second from c2.

Important note: The following requirements refer to the same project. The requirements are described separately for clarity's sake. You are advised to read both before you begin your work.

# Requirement A: Developing a generic software

We would like to implement the clustering algorithm in a generic way, so that it supports items implementing the following interface:

**public** **interface** Clusterable <T>{

**public** **double** distance(T other);

}

In particular, we would like to use the clustering algorithm for two different data types:

1. Points on a two-dimensional plane
2. Strings of n bits

For two-dimensional points we will implement the following class:

**public** **class** TwoDPoint **…** {

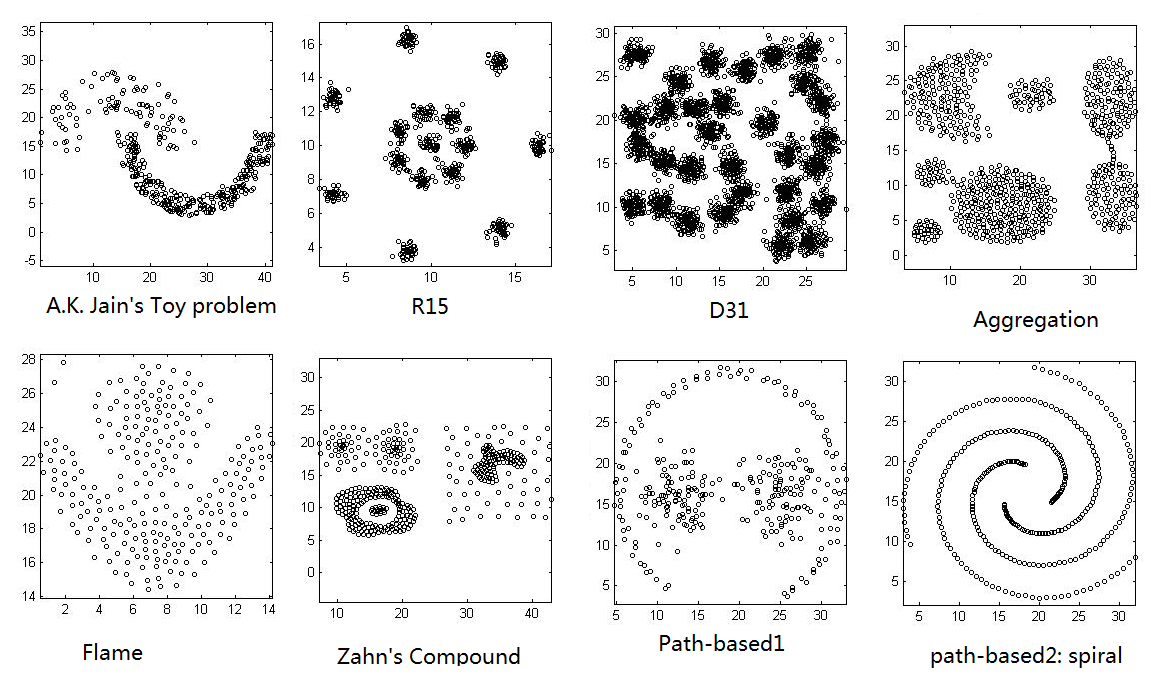
**double** x;

**double** y;

}

For TwoDPoint, use the [Euclidean distance](https://en.wikipedia.org/wiki/Euclidean_distance).

For your tests, you can use the [spiral dataset](http://cs.joensuu.fi/sipu/datasets/spiral.txt) (depicted below). Find what threshold works for separating the dataset to 3 spirals.



For bit-strings we will implement the following class:

**public** **class** BitArray **… {**

ArrayList<Boolean> bits;

…

}

For BitArray, use the [hamming distance](https://en.wikipedia.org/wiki/Hamming_distance)

Advice: To avoid bugs, when updating a Set-element (for example when you replace two clusters with their union), first remove the element from the Set, update it and then return it into the Set. This will prevent inconsistencies between the element contents and its hash key.

# Requirements, Part B: Using Java Streams

**Important requirement: The usage of loops is prohibited throughout this project. Any iterative operation on the data structures should be implemented using streams only. In addition, you must not use Stream's foreach function.**

**The *while* loop in the supplied algorithm is the only exception to the requirement prohibiting loops.**

Advice: If you find it difficult to write a loop-free code, start by writing and verifying your code with loops, then "convert" each code section involving loops to streams-based code. You can find [here](https://stackoverflow.com/questions/40678892/replace-nested-loop-with-java-8-flatmap) an example for converting a nested loop to a Java 8 streams expression. Add functions to make your code modular.

In addition, you may find the [AbstractMap.SimpleEntry](https://docs.oracle.com/javase/7/docs/api/java/util/AbstractMap.SimpleEntry.html) class useful for representing pairs of objects.

You are provided a skeleton of the code, a sample set of two-dimensional points, a sample set of bitarrays and their corresponding clustering. You should complete the classes AgglomerativeClustering, BitArray, TwoDPoint and submit and verify them through the VPL component. You are also required to fill the file with the submitters' names and IDs. Change the code according to the TODO comments in the code.

**The provided skeleton**

Good luck!