7th Assignment – Graphs: Minimum spanning tree

Instructions

- Download file **cal_fp07_CLion.zip** from the course's Moodle area (it contains folder **lib,** folder **Tests** with files **tests.cpp** and **Graph.h**, and files **CMakeLists** and **main.cpp**, following up previous practicals)
- Open a ***project*** in CLion, and select the folder containing the files as described above
- Do "*Load CMake Project*" over the file *CMakeLists.txt*
- Run the project (**Run**)
- Note that ***unit tests in this project may be commented***. If this is the case, then uncomment the tests as you make progress in the implementation of your solutions
- ***You should follow the order of exercises in this practical class***
- Implement your solution in the respective ***.cpp*** files, in case you're not implementing a template. Templates must be implemented in the respective ***.h*** files
- Important note: If you need access to external files in I/O mode, you should set up their location by defining the CLion IDE's "Working Directory" environment variable, selecting it from the menu Run > Edit Configurations… > Working Direcory
- The parts to be coded in file **Graph.h** are marked with **TODO** and may include comments and hints on how to implement them

Exercises

Consider the **Graph** class you used in previous classes, which is defined in the *Graph.h* file. You should edit the classes in *Graph.h* in order to complete the exercises below. Look at the *tests.cpp* file in order to identify auxiliary functions which are required but are not explicitly asked for.

a) Implement the following method in the **Graph** class:

```
vector<Vertex<T>*> calculatePrim()
```

This function implements Prim's algorithm to find the minimum spanning tree from the first vertex ***v*** in the graph, to all other vertices.

b) Implement the following method in the **Graph** class:

```
vector<Vertex<T>*> calculateKruskal()
```

This method implements Kruskal's algorithm to find the minimum spanning tree.