Continental

Algorithmic Task (C++)

(1) You will be given 2 polylines as std::vector of points. Write a function in C++11, that checks if the two polylines are closer to each other than a given distance threshold. Return true if they are closer, and false if they are not. The distance threshold is 1.5F.

Please implement your code in a function like this:

```cpp
bool arePolylinesCloserThanThreshold(std::vector<sPoint2D>& polyline1,
std::vector<sPoint2D>& polyline2)
{
  // Implement your code here (C++11)!
  // ...
  return closerThanThreshold;
}
```

Example input:
```cpp
std::vector<sPoint2D> polyline1 { sPoint2D(2.0F, 3.0F), sPoint2D(3.0F, 4.0F),
sPoint2D(2.0F, 6.0F) };
std::vector<sPoint2D> polyline2 { sPoint2D(5.0F, 6.0F), sPoint2D(5.0F, 4.0F),
sPoint2D(7.0F, 4.0F), sPoint2D(7.0F, 2.0F) };
```

Expected output: false, because the minimum distance is 2.0F.

The useable point struct is:
```cpp
struct sPoint2D
{
  sPoint2D(float xValue, float yValue)
  {
    x = xValue;
    y = yValue;
  }
  float x;
  float y;
};
```

(2) Analyze the runtime complexity of your algorithm. How could it be further improved?

(3) Create a simple cmake project that creates a library from the above function.

(4) With a test framework of your choice (catch2, googletest, etc.) test the function from (1.) and include the tests in the cmake project from (3.).

Please send your solution in an e-mail attachment. We kindly ask you not to share your solution on any public website (e.g. github.com, gitlab.com, medium.com, etc.).

The ZIP should contain only the source code, the build-configuration files, which are required to build your solution, a short description how to build your solution and optionally other documentation, which you would like to add. You don't need to include the source code of the 3rd party testing library (e.g. googletest), which you chose, if it's too large for an email. But in this case, please indicate the version and/or git hash of the library you used.

This document is the property of Continental. It may not be distributed or reproduced in any form.