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:

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

Example input:

```
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:

```
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?