2circles- solution
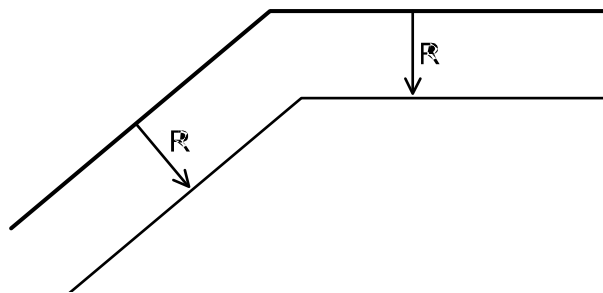
The idea is to do a binary search for the radius - R. For a fixed radius we "deflate" the polygon by R, thus also reducing the circles to single points. In this case we only have to find the furthest 2 points in the new polygon. If these two points are more than 2 * R apart then R is a possible answer and we should search for a higher R, otherwise we should search for a lower R. For each value of R there are two sub problems:

Sub problem I – Deflating the polygon:

For a radius R we shift each segment towards the center of the polygon with R like in the figure below. The secret is that the new line will be parallel to the old segment.

The next step is to find the intersection of these lines and obtain the new polygon. By this process some of the edges may disappear.

O(N^2) solution: we consider a square that contains the polygon and then intersect it with each of the shifted lines of the old polygon.

O(N) solution: we make use of the fact that the points are in trigonometric order. We will keep the segments in a stack. Initially the stack contains the first shifted edge (2 points). When trying to introduce a new edge we must first check if the previous edge is not outside the polygon, otherwise it is eliminated. When an old edge is not entirely outside the new edge we replace the last point on the stack with the intersection between the two and add another point to the stack, the end of the new edge.
After introducing all edges we must also check the beginning of the stack since some edges from the beginning might have been eliminated by the last line.

Sub problem II – Finding the two furthest points:

O(N^2) solution: taking the distance between all pairs and selecting the maximum distance.

O(N) solution: For the diameter problem there exists a classic algorithm called rotating calipers in O(N) time. An explanation can be found here http://cgm.cs.mcgill.ca/~orm/rotcal.html.

Putting things together

Thus the final solution is O(NlogMAX), where MAX is the maximum possible radius. A good upper bound for the binary search is D/4 where D is the maximum distance between two points of the initial polygon.

The O(N^2logMAX) solution should obtain 50 points.

For further reference you can check:
http://www.springerlink.com/content/yv8ju0fl60823whx/ for an O(N) solution.