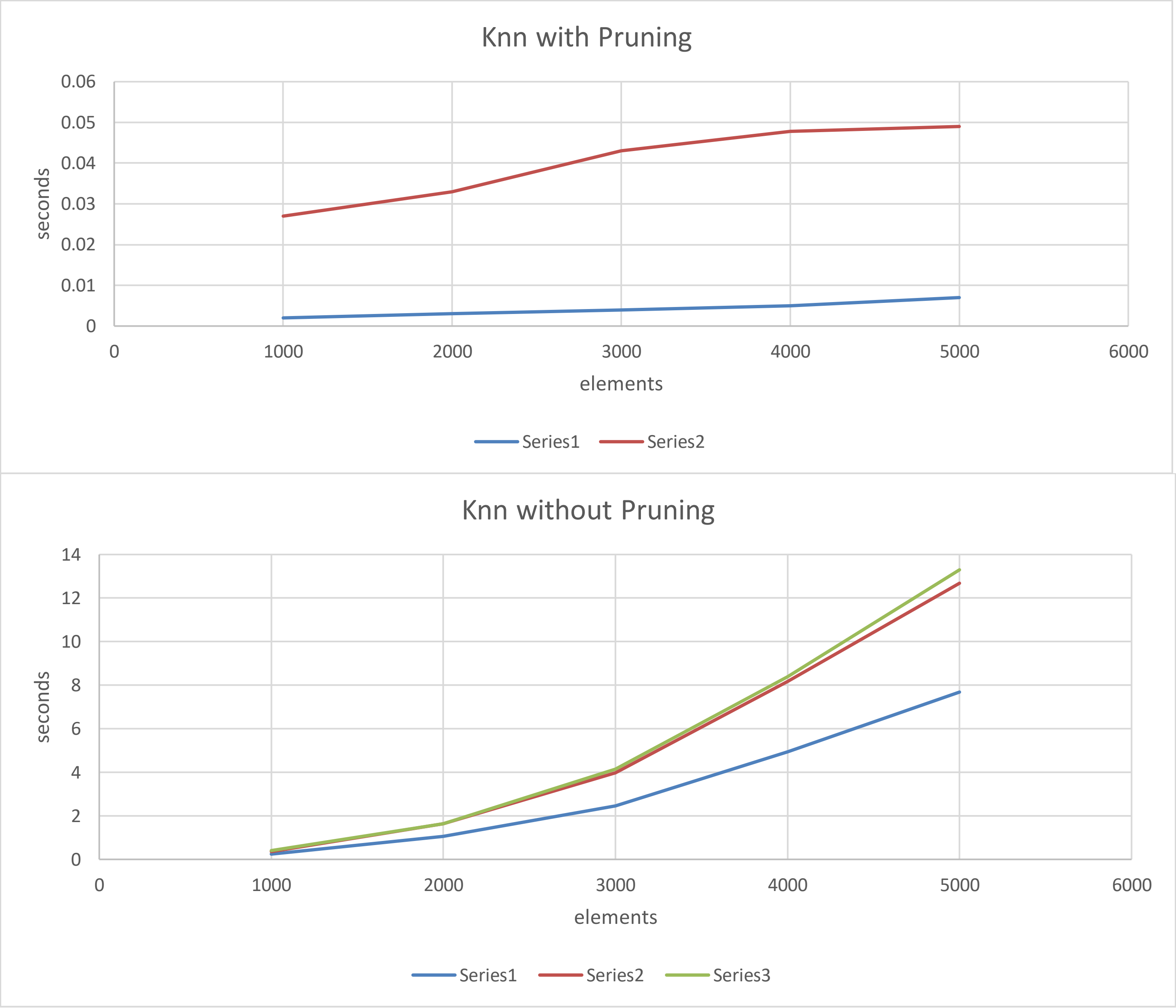
K-D : BLUE
QUAD : RED



As we can see we implemented to the code a pruning logic and the results are so much better. At the K-D we compare the two euclidian distances of the two childs(right and left) of the node and we compare them to the last euclidian distance of the list we keep the distances which is sorted every time. So if the distance of one child is smaller then we use recursion for this child and the distance that was compared to is replaced by the new distance. So its time we compare the distance we want to add with the biggest of the list. In the Quad it's the same thing but now we have four childs each time, and we have to do four compares. If the distances are smaller, then we do recursion for those nodes and so on.