Convex Hull

1. (2 points) Given a list of points, develop a linear time algorithm that tests whether the points form a convex hull or not. Notice that the points might come in either CW or CCW order. In both cases, the algorithm should return true as long as they form a convex hull.

Example 1: Input: 
\[
[(0,0), (1,0), (2,2)]
\] ➔ Output: True

Example 2: Input: 
\[
[(0,0), (2,0), (1,1), (1,3)]
\] ➔ Output: False

Example 3: Input: 
\[
[(0,0), (1,1), (2,2), (1,0)]
\] ➔ Output: True

2. (3 points) Given a set of points \( P \) and a straight line defined by two points \( p_1 \) and \( p_2 \), prove that the point \( p_i \in P \) that is farthest away from the line \( 
\overline{p_1p_2} \) is part of the convex hull of \( P \). This can also be expressed using the following mathematical expression.

\[
p_i \in P \land p_j \in P \land dist(p_i, \overline{p_1p_2}) \geq\ dist(p_j, \overline{p_1p_2}) \Rightarrow p_i \in CH(P)
\]

where \( dist(p, \overline{p_1p_2}) \) is the Euclidean distance between the point \( p \) and its projection \( p^* \) on the line \( \overline{p_1p_2} \) and \( CH(P) \) is the convex hull of \( P \).

This proof is needed for the recursive part of the Quick Hull algorithm.

3. (2 points) Building on your proof in 2, prove that the farthest pair of points \( p_1 \) and \( p_2 \) in a set \( P \) have to be both on the convex hull of \( P \). This can be expressed using the following expression.

\[
p_{i,j,k,l} \in P \land dist(p_i, p_j) \geq dist(p_k, p_l) \Rightarrow p_{i,j} \in CH(P)
\]

where \( p_{i,j,k,l} \in P \) means that all of the points \( p_i, p_j, p_k, \) and \( p_l \) are in the set \( P \).

4. (3 points) Describe how to craft a worst-case input of size \( n \) points for the Quick Hull algorithm. Recall that a worst-case scenario of the Quick Hull algorithm yields an \( O(n^2) \) asymptotic running time.