Q) Divide and conquer algorithm to return the set of all corner points.

```
cornerPoints (P, left, right):
    if |P| ≤ 1 : return P
    m = medianPoint (P)
    L = cornerPoints (P[left ... m], left, m)
    R = cornerPoints (P[m+1 ... right], m+1, right)
    i = 1
    l = L[i]
    r = R[i]
    // consider the points in L in decreasing order of y coordinate
    while ( i < |L| )
        if ( l.y ≤ r.y )
            remove l from L
        i = i + 1
        l = L[i]
    Q = all unique points in L and R
    return Q
```

The time complexity of cornerPoints() is as follows:

$$T(n) = \begin{cases} 2T(n/2) + cn & \text{if } n \geq 2 \\ c & \text{if } n < 2 \end{cases}$$

using master's theorem we can easily obtain

$$T(n) = \Theta(n \log n)$$

Proof of Correctness

Base case: If size of set P is equals to 1, then only one corner point exits.

Induction Hypothesis: Assume that cornerPoints(P[l...m], l, m) will give all the corner points in the left partition of the 2D plane, and cornerPoints (P[m+1...h], m+1, h) will give all the corner points in the right partition of the 2D plane, where the partition is imaginarily assumed with respect to a median point $p \in P$.

Induction Claim: To show that cornerPoints (P[l...h], l, h) will give all the corner points in the 2D plane.

Proof: By IH, L will contain all the corner points in left partition w.r.t median point and R will contain all the corner points of the right partition. Now there might be some points in L which have a point above/right of it in the right partition. Hence viewing from those points of the right partition R, few points of L may not be considered as corner points anymore. Thus, a point p in L will not be a corner point of the whole 2D-plane, if there exists a point q in R, such that; $p.y \leq q.y$ (i.e, y-component of p is less than that of q) Such points are removed from L, by comparison in the while loop in line 10. Now all of the corner points of L are being compared with the point in R which has highest y component. If any point in the left partition L lies below or left w.r.t any point in R, then obviously such points of L can be removed using the 1st point of R only, as it has the highest y-component (points in R considered in decreasing order of y-component). Thus we will get all corner points of the 2D plane from (L & R).