Time Spent: 2 hours

Collaborators and Resources:

Problem 1 (a)

For part (a), I want to first argue that for any line segment L connecting points p and q in S, the segment L is on the boundary of the extreme envelope of S if and only if all other points in S are on one side of L. This is true because any extreme envelope is going to be a convex polygon. Furthermore by definition all points in S are going to be contained within this polygon. If all points in S are not on the same side of L, then the polygon will not contain all points ...

In order to test which side of L a given point r = (x_3,y_3) is on, we basically need to calculate the equation for line L which is given in point-slope form: given two points (x_1,y_1) and (x_2,y_2) we know that y - y_1 = ((y_2-y_1)/(x_2-x_1))*(x-x_1) $\implies$ y = ((y_2-y_1)/(x_2-x_1)) $\cdot$ (x-x_1) + y_1. In the special case that the line L is vertical (there is no slope) we need to just say that if $x_3 < x_1$=x_2 then r is on the left and if $x_3 > x_1$=x_2 then r is on the right. There is no else case because there are no three co-linear points.
If the line has a slope then we can just check if $y_3 > y$ (y_3 is above the line L) or if $y_3 < y$ (y_3 is below the line L).

(b)

In this case we're going to find the set of points E$\subseteq$S on the boundary of the extreme envelope of S. For each pair of points we need to test each line segment ...

Given a set of points S in the plane:

1. Initialize E = $\emptyset$
2. For each pair of points p = (x_1,y_1) and q = (x_2,y_2) in E:
   (a) For points r = (x_i,y_i) in E\{p,q}:

i. If $x_1 = x_2$:
   A. If $x_i < x_1$ $\forall$x$_i$ or $x_i > x_1$ $\forall$x$_i$ then add p, q to E
ii. Else:
   A. Initialize y = $((y_2\text{-}y_1)/(x_2\text{-}x_1)) \cdot (x_i\text{-}x_1) + y_1$
   B. If all $y_i > y$ or all $y_i < y$ then add p, q to E

Claim 1. The algorithm determines all points which are on the boundary of the extreme envelope of S.

Proof. For all points p and q, my algorithm will check if the line formed by p, q is vertical. If so, for all other points r it will check if they are all on the right side or are all on the left side of the line segment. If the line has a slope then we will make the same comparisos but checking whether the points are above or below the line L ... □

Claim 2. The algorithm has O(n^3) worst-case running time.

Proof. The reason for this is that there are n^2 possible pairs and checking each pair could take O(n^2) time. Furthermore there are going to be n-2 points r remaining after selecting a pair, and so this will effectively require another n-2 comparisons making it O(n^3) time. Adding the points in the pair to the set of boundary points and checking if this should be done are both operations that can be done in linear time and so will not affect the overall worst-case time of O(n^3) ... □