



Capstone 2019

Visualizing Convex Hulls

Michael Schirger



Capstone 2019

Project Assignment

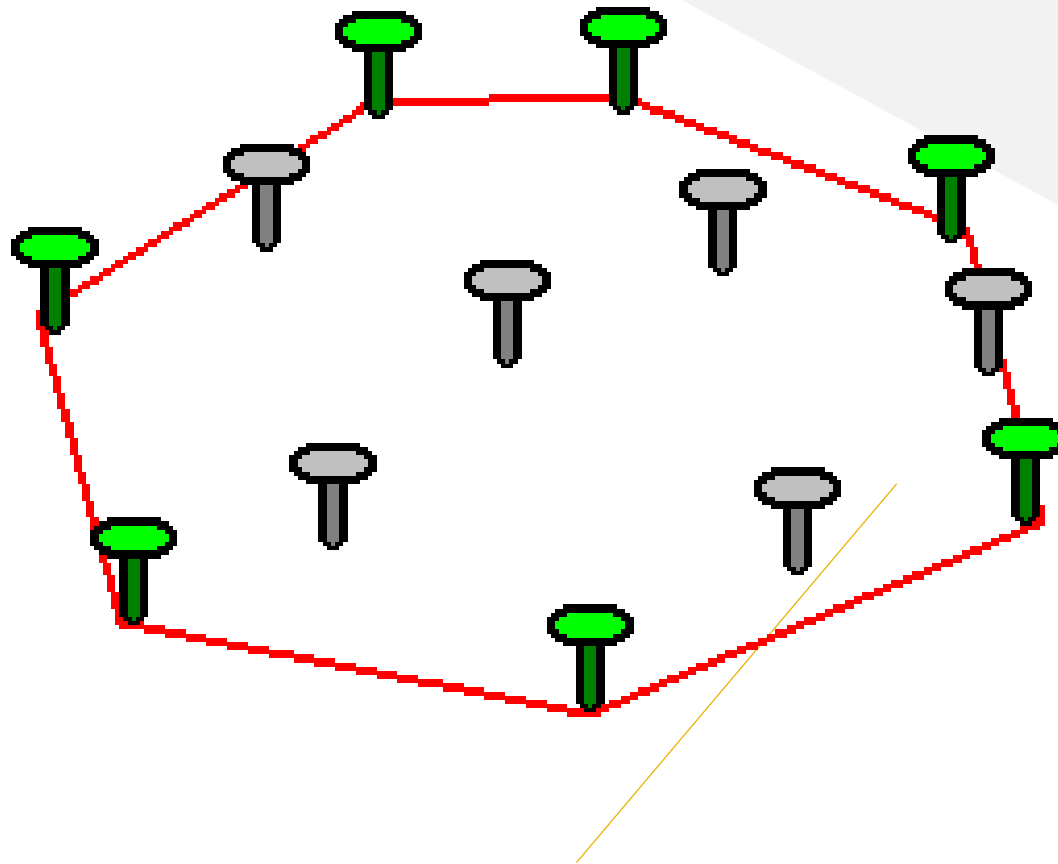
What I was given

Project Assignment

- Develop an app that collects GPS coordinates and then displays the convex hull of the set of locations as an overlay on a map
 - GPS coordinate collection
 - Collects and stores coordinates for a specified amount of time.
 - Continues to collect if the app is in the background.
 - Construct the Convex Hull
 - Design and implement an algorithm to construct convex hull.
 - Illustrate convex hull as an overlay on a map
 - Use different colors for different collections.
 - Show coordinates of vertices when requested.
 - Show intersection or union of N convex hulls.
 - Compute the area of a convex hull when requested.
 - Application?

Convex Hull

- Smallest convex polygon that contains all the points of the set.
- A line drawn between any two points must remain within the convex hull.





Capstone 2019

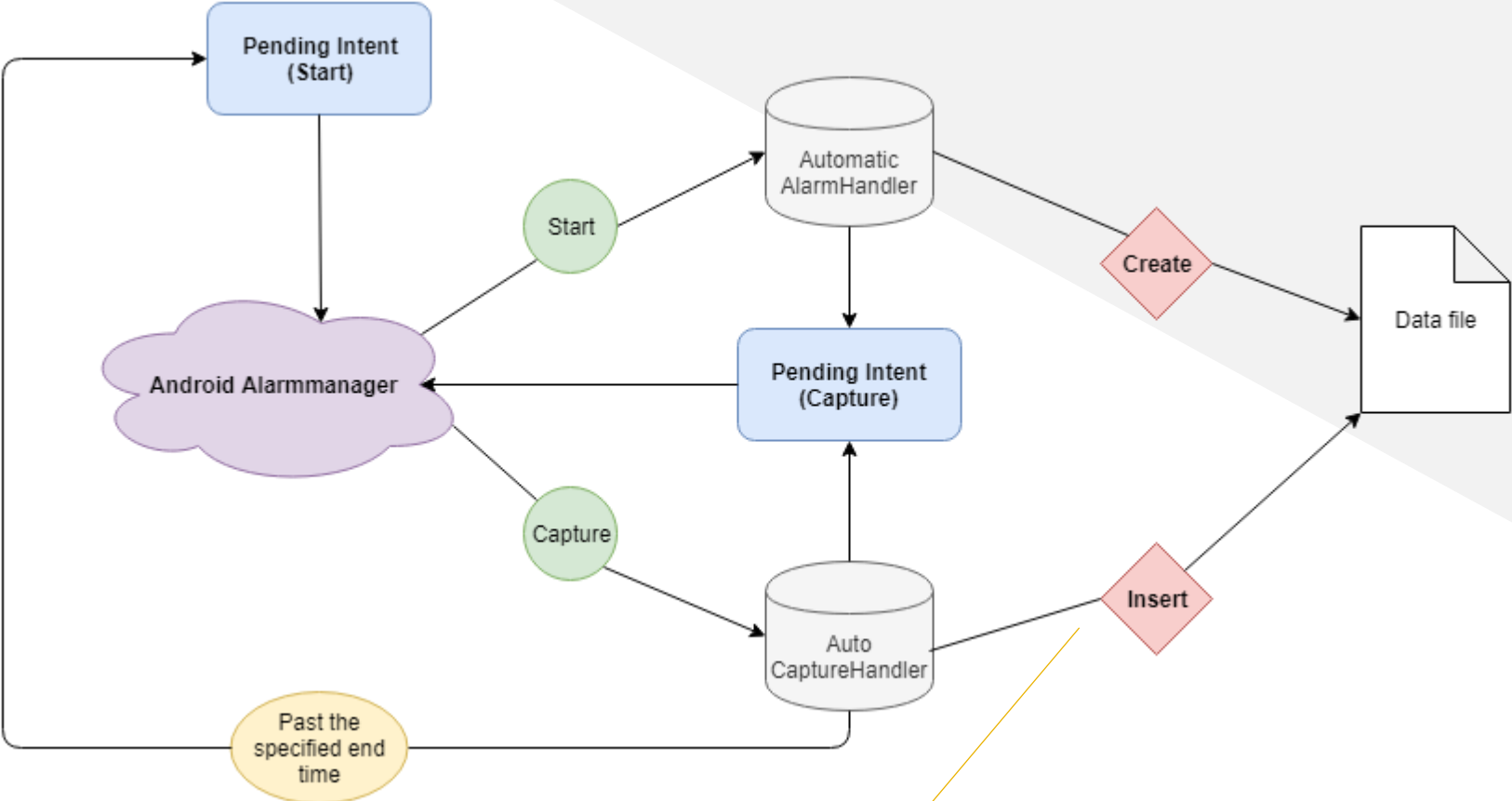
Part 1

GPS Coordinate Collection

Coordinate Collection

- The user is given four options for coordinate collection
 - Automatic Start
 - Set a start and end time and the app will capture coordinates within that time period.
 - This is the most useful method.
 - Manual Start
 - Start it by pressing a button and have it record for a specified amount of time.
 - Full Manual
 - Start it by pressing a button and have it record until the end button is pressed.
 - Single coordinate captured
 - Will capture a single coordinate and place it into chosen file.

Coordinate Collection



Coordinate Collection

- Android documentation recommends the use of *FusedLocationProviderClient*
 - This simply retrieves the latest location in the location cache.
 - Will not work for background location capture.
- For background capture it is necessary to use *LocationServices*
 - Make a call to *LocationServices* API.
 - Explicitly requests current location of device.
 - *LocationServices* uses a callback function
 - Returns latitude and longitude of device to the callback function.
- Once the location has been retrieved, the latitude and longitude are written to a data file.



Capstone 2019

Part 2

Construct the Convex Hull

Data Interpretation

- In terms of standard Cartesian coordinate system
 - Longitude is the x value.
 - Latitude is the y value.
 - (x,y) becomes (longitude, latitude).
- So GPS coordinates can be interpreted as Cartesian coordinates.
- This is important for the construction of convex hulls.

Algorithm

- Andrew's Monotone Chain Convex Hull Algorithm

- Takes in a sorted arraylist of coordinates
 - Sorted based on the x (longitude) value

- Iterates through the arraylist twice and determines the vertices using

$$[(q.x - r.x) * (p.y - r.y)] \geq [(q.y - r.y) * (p.x - r.x)]$$

for points q, p, and r.

- Constructs the convex hull in two phases
 - Upper Convex Hull
 - Lower Convex Hull

Algorithm

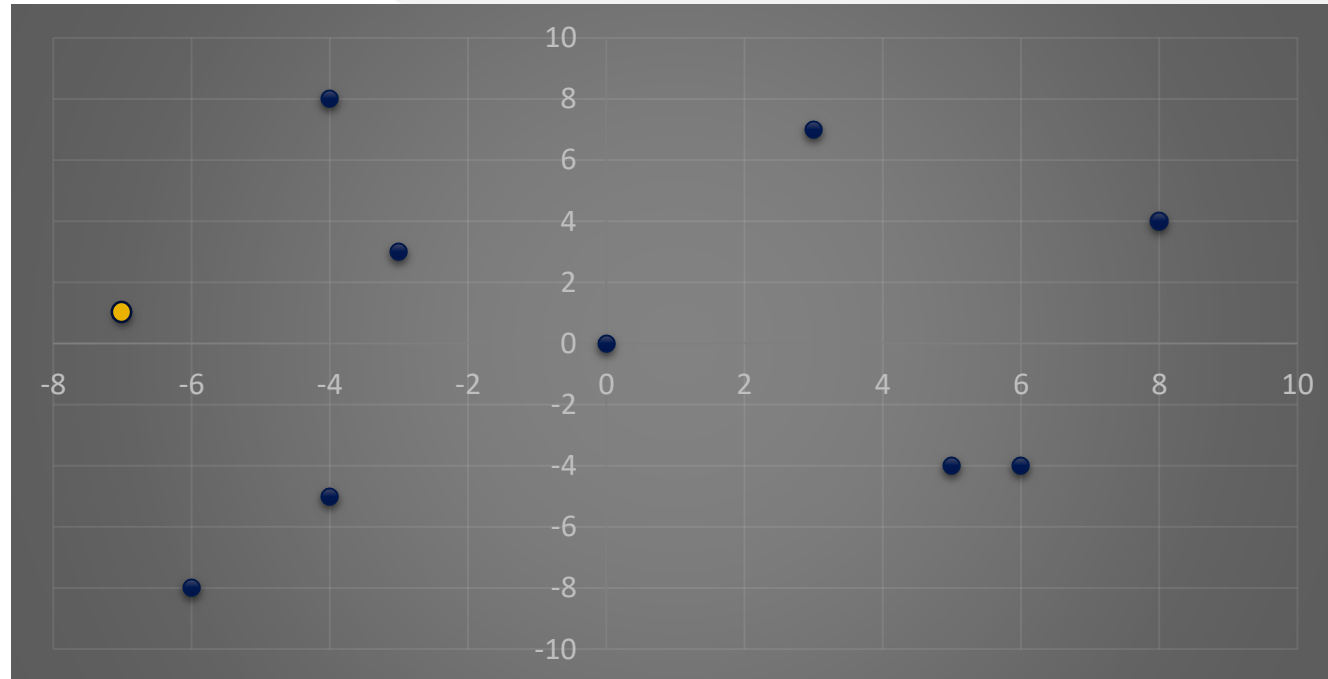
- If there are points

$(0,0)$, $(-4,8)$, $(-4,-5)$, $(3,7)$, $(8,4)$,
 $(5,-4)$, $(-7,1)$, $(-6,-8)$, $(-3,3)$, $(6,-4)$

The algorithm will begin by sorting the points in an arraylist:



Algorithm



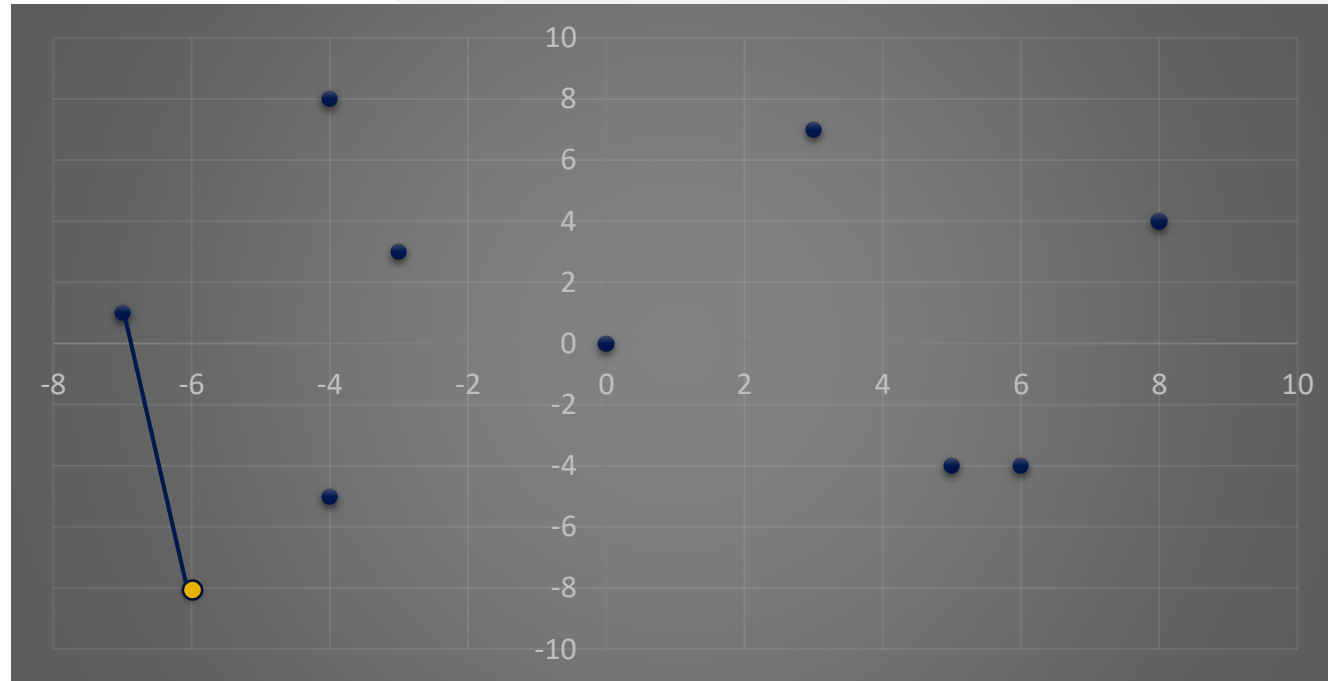
Coordinates:



Vertices:

Check:

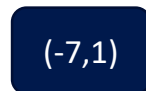
Algorithm



Coordinates:

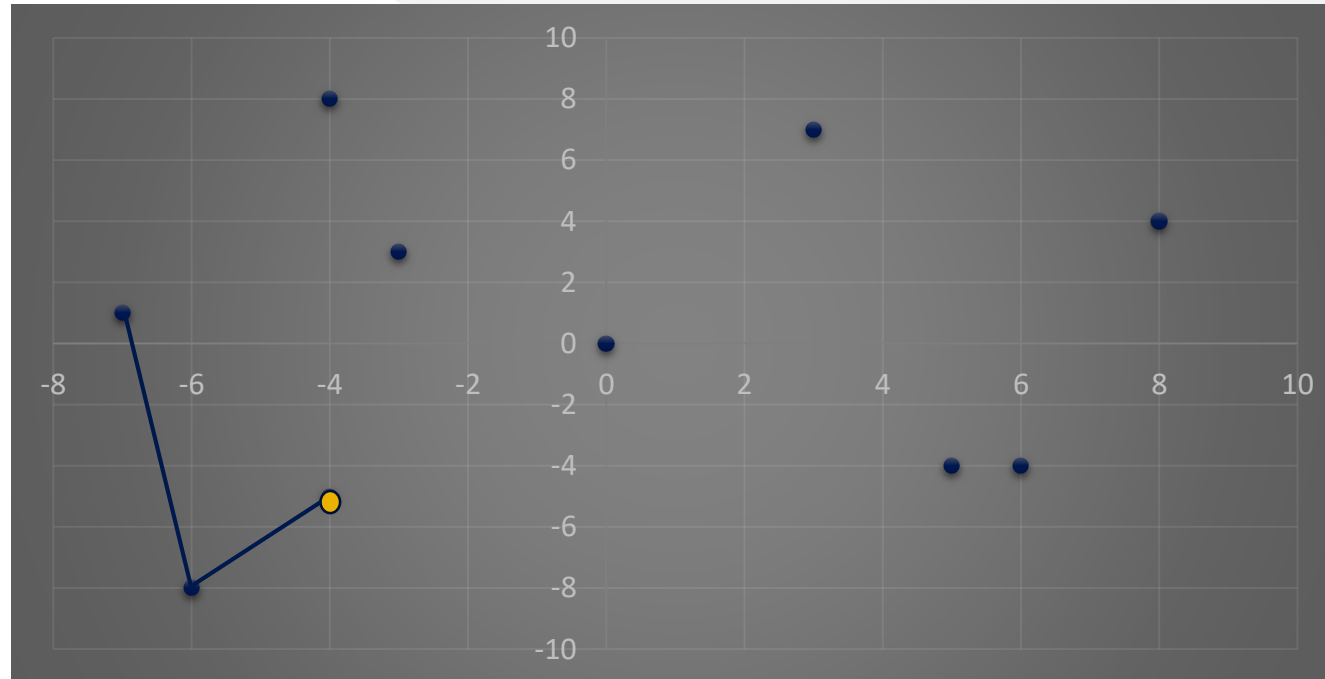


Vertices:



Check:

Algorithm

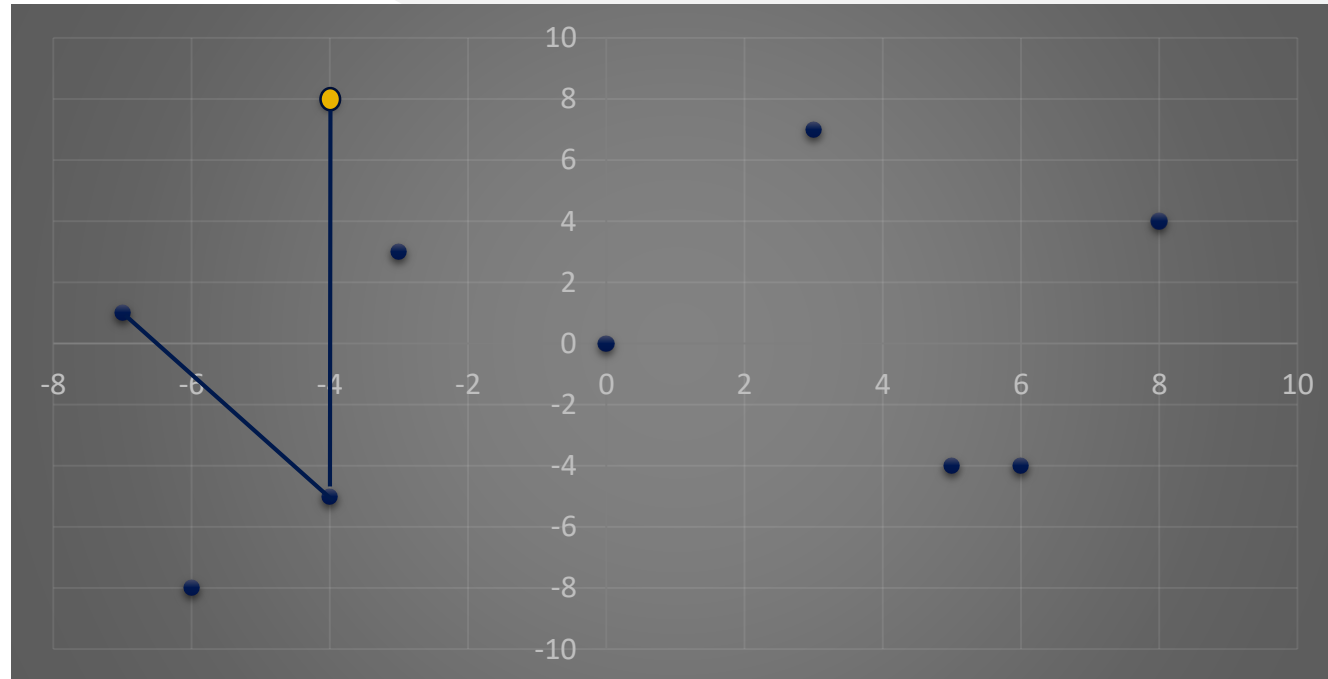


Coordinates: $(-7,1) \rightarrow (-6,-8) \rightarrow (-4,-5) \rightarrow (-4,8) \rightarrow (-3,3) \rightarrow (0,0) \rightarrow (3,7) \rightarrow (5,-4) \rightarrow (6,-4) \rightarrow (8,4)$

Vertices: $(-7,1) \rightarrow (-6,-8)$

Check: $p: (-4,-5) \ q: (-6,-8) \ r: (-7,1) \rightarrow -6 > -27$

Algorithm

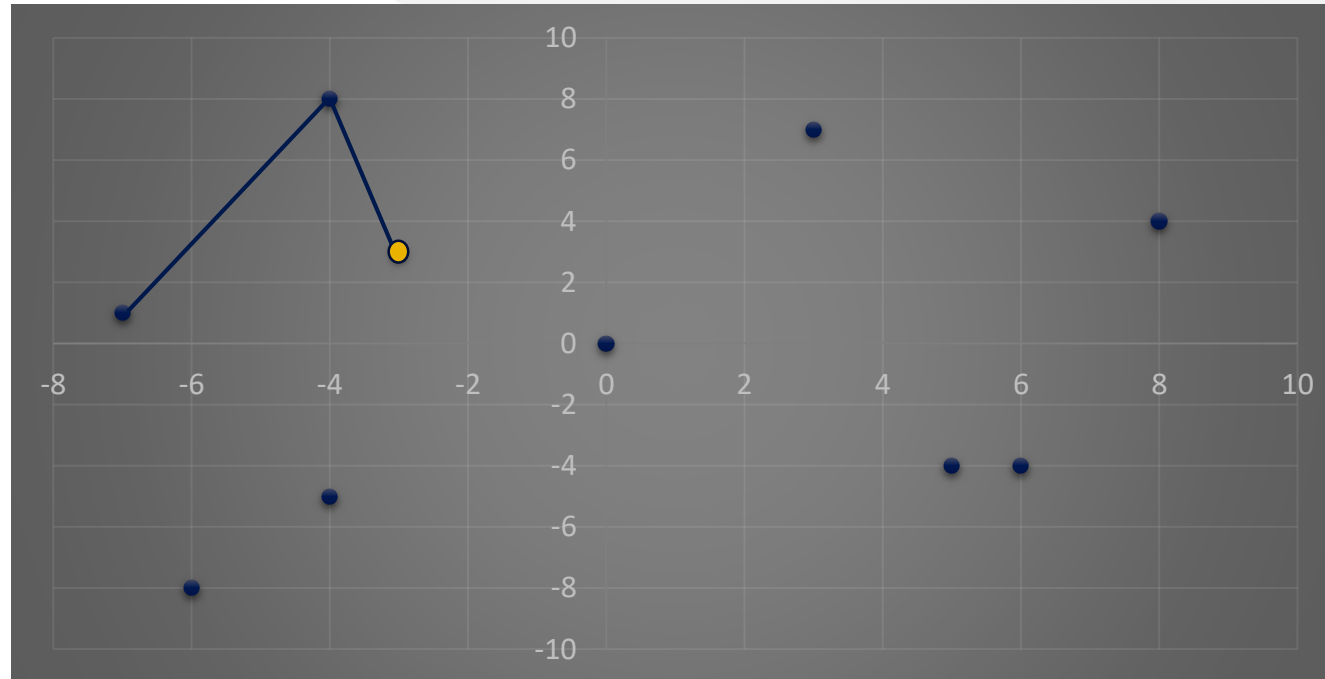


Coordinates: $(-7,1) \rightarrow (-6,-8) \rightarrow (-4,-5) \rightarrow (-4,8) \rightarrow (-3,3) \rightarrow (0,0) \rightarrow (3,7) \rightarrow (5,-4) \rightarrow (6,-4) \rightarrow (8,4)$

Vertices: $(-7,1) \rightarrow (-4,-5)$

Check: $p: (-4,8) \ q: (-4,-5) \ r: (-7,1) \Rightarrow 21 > -18$

Algorithm

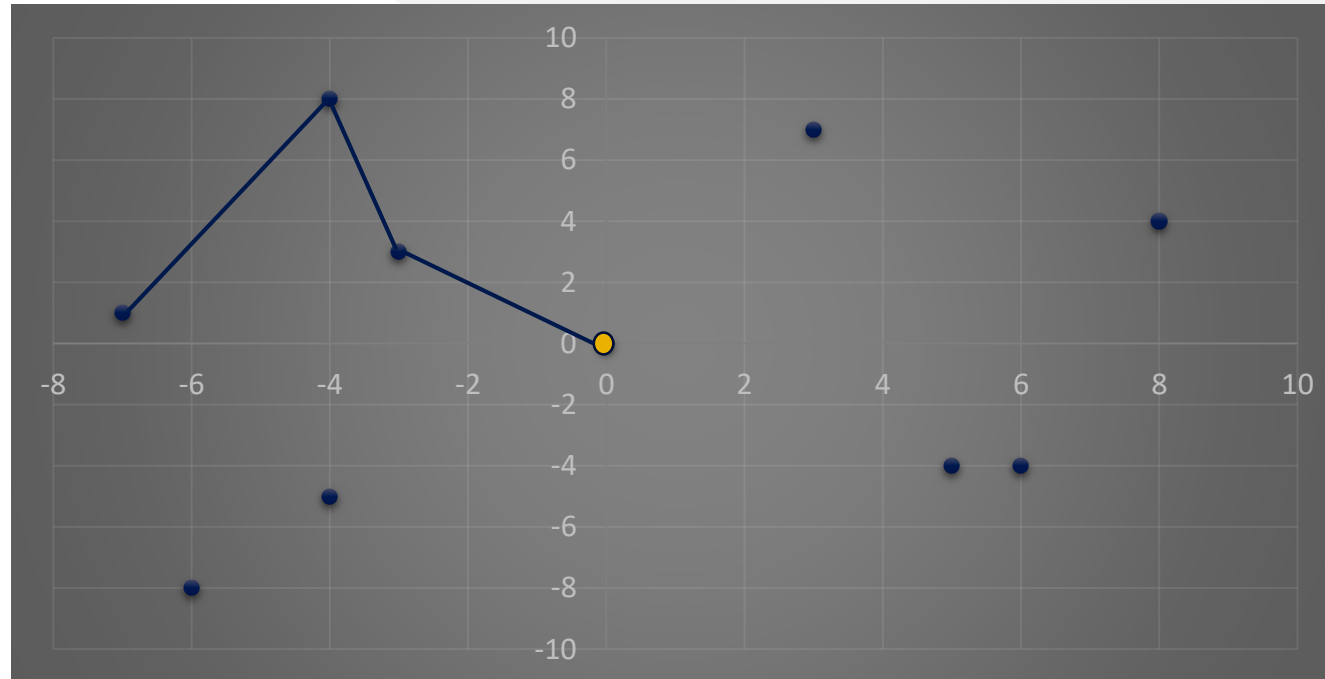


Coordinates: $(-7,1) \rightarrow (-6,-8) \rightarrow (-4,-5) \rightarrow (-4,8) \rightarrow (-3,3) \rightarrow (0,0) \rightarrow (3,7) \rightarrow (5,-4) \rightarrow (6,-4) \rightarrow (8,4)$

Vertices: $(-7,1) \rightarrow (-4,8)$

Check: $p: (-3,3) \ q: (-4,8) \ r: (-7,1) \ \rightarrow \ 6 < 28$

Algorithm

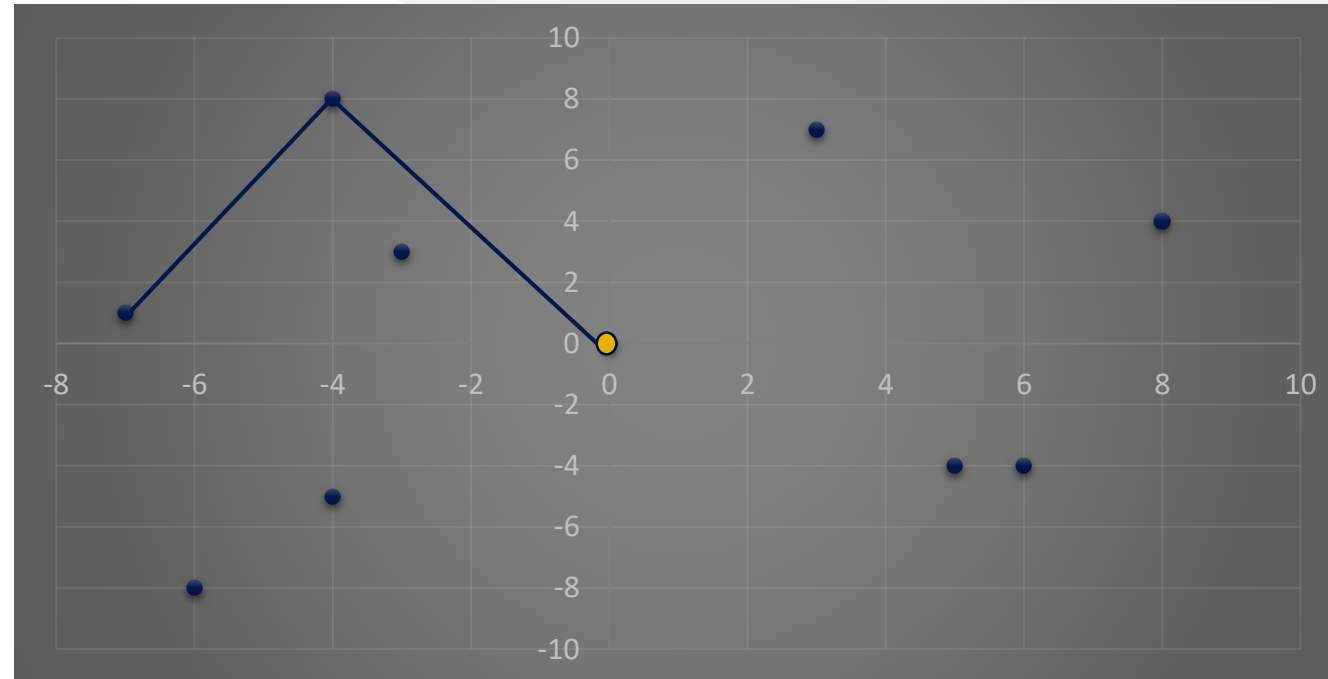


Coordinates: $(-7,1) \rightarrow (-6,-8) \rightarrow (-4,-5) \rightarrow (-4,8) \rightarrow (-3,3) \rightarrow (0,0) \rightarrow (3,7) \rightarrow (5,-4) \rightarrow (6,-4) \rightarrow (8,4)$

Vertices: $(-7,1) \rightarrow (-4,8) \rightarrow (-3,3)$

Check: $p: (0,0) \ q: (-3,3) \ r: (-4,8) \rightarrow -8 > -20$

Algorithm

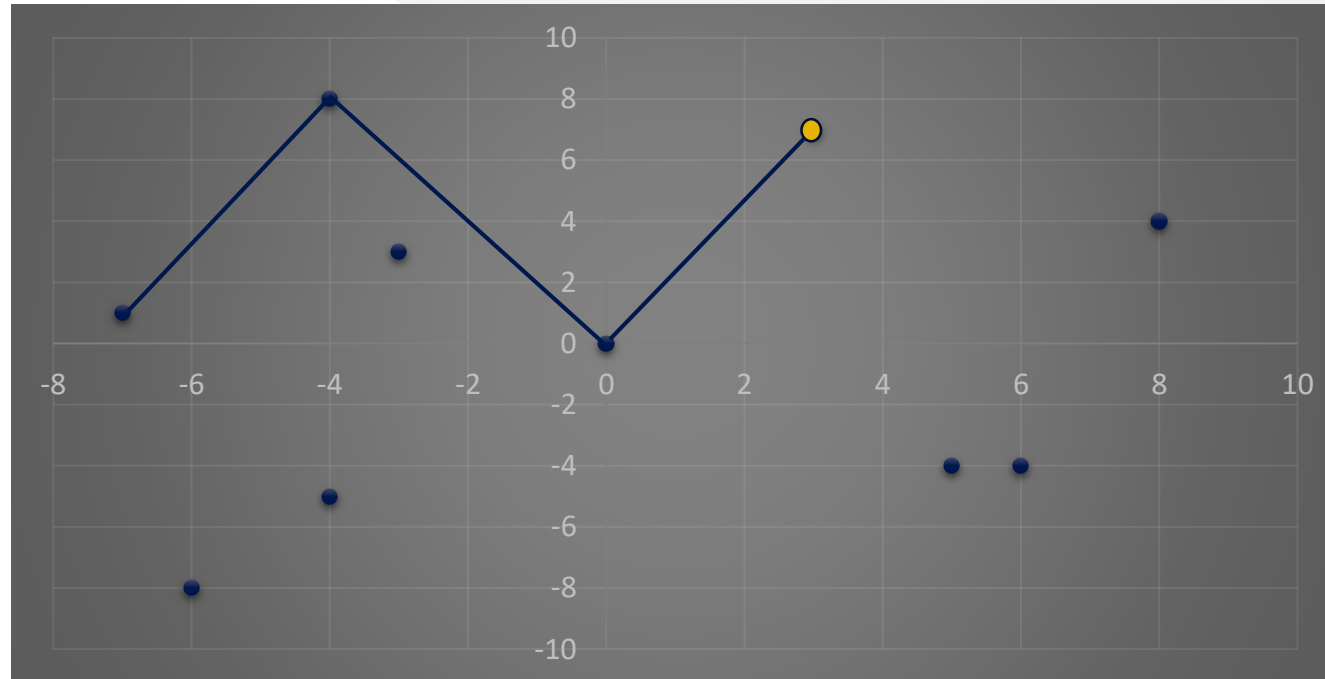


Coordinates: $(-7,1) \rightarrow (-6,-8) \rightarrow (-4,-5) \rightarrow (-4,8) \rightarrow (-3,3) \rightarrow (0,0) \rightarrow (3,7) \rightarrow (5,-4) \rightarrow (6,-4) \rightarrow (8,4)$

Vertices: $(-7,1) \rightarrow (-4,8)$

Check: $p: (0,0) \ q: (-4,8) \ r: (-7,1) \rightarrow -3 < 49$

Algorithm

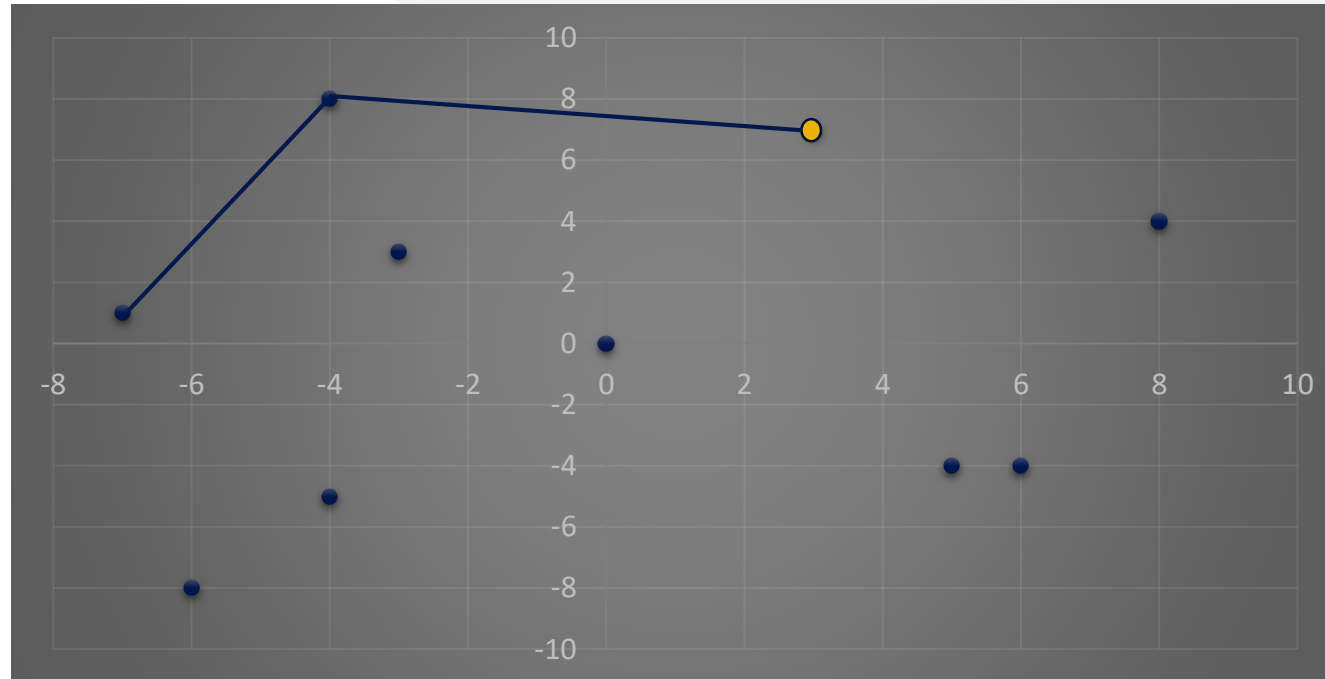


Coordinates: $(-7,1) \rightarrow (-6,-8) \rightarrow (-4,-5) \rightarrow (-4,8) \rightarrow (-3,3) \rightarrow (0,0) \rightarrow (3,7) \rightarrow (5,-4) \rightarrow (6,-4) \rightarrow (8,4)$

Vertices: $(-7,1) \rightarrow (-4,8) \rightarrow (0,0)$

Check: $p: (3,7) \ q:(0,0) \ r:(-4,8) \ \rightarrow \ -4 > -56$

Algorithm



Coordinates:



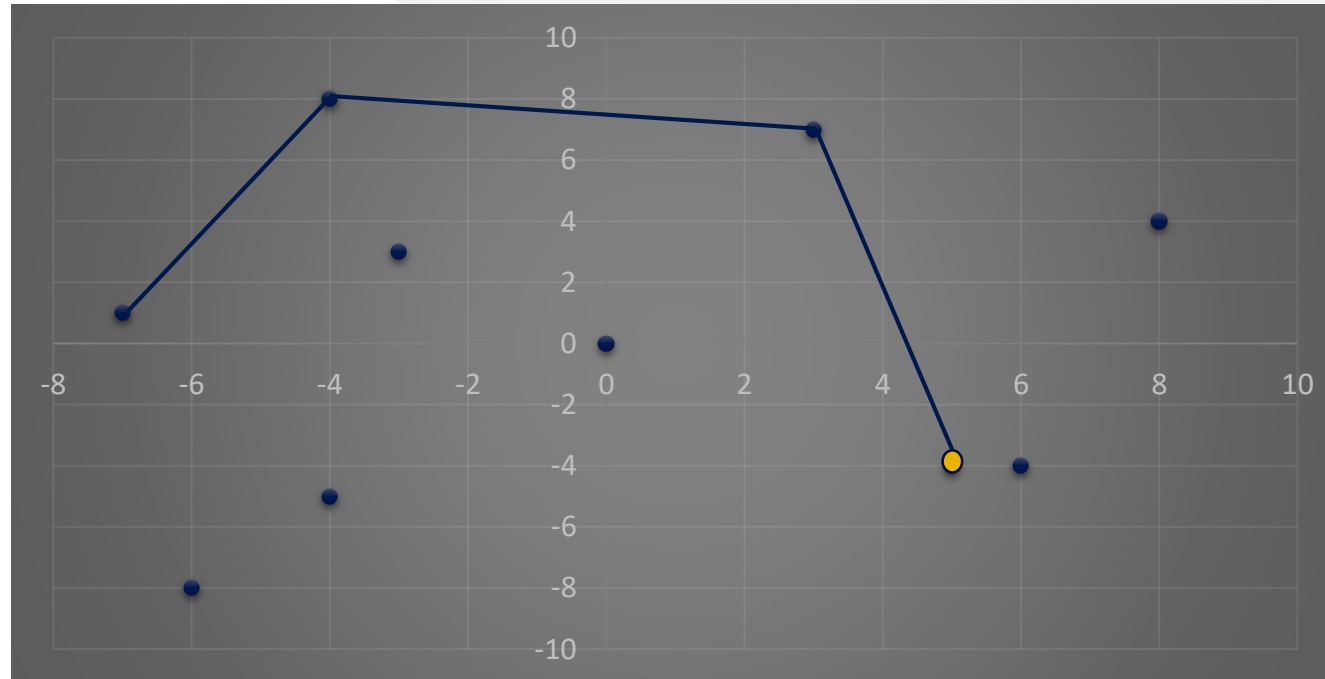
Vertices:



Check:

$p: (3,7) \quad q: (-4,8) \quad r: (-7,1) \quad \Rightarrow \quad 0 < 70$

Algorithm

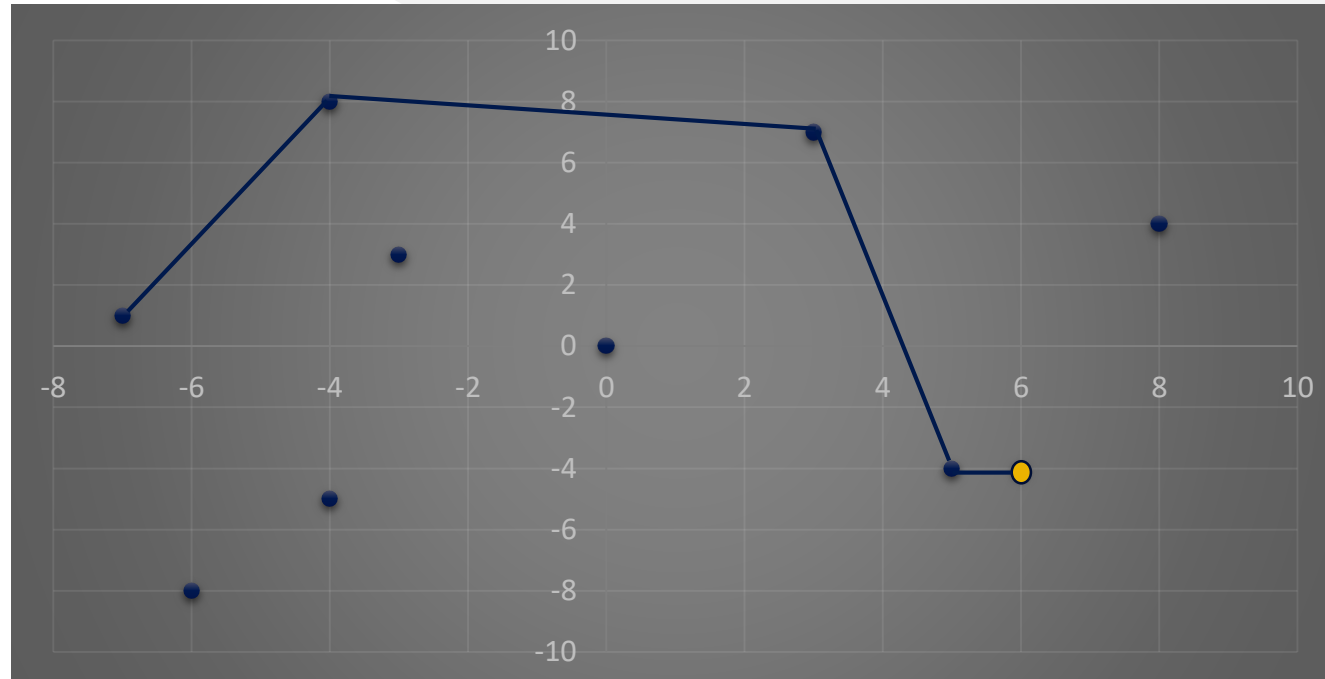


Coordinates: $(-7,1) \rightarrow (-6,-8) \rightarrow (-4,-5) \rightarrow (-4,8) \rightarrow (-3,3) \rightarrow (0,0) \rightarrow (3,7) \rightarrow (5,-4) \rightarrow (6,-4) \rightarrow (8,4)$

Vertices: $(-7,1) \rightarrow (-4,8) \rightarrow (3,7)$

Check: $p: (5,-4) \ q:(3,7) \ r:(-4,8) \ \rightarrow \ -84 < -9$

Algorithm

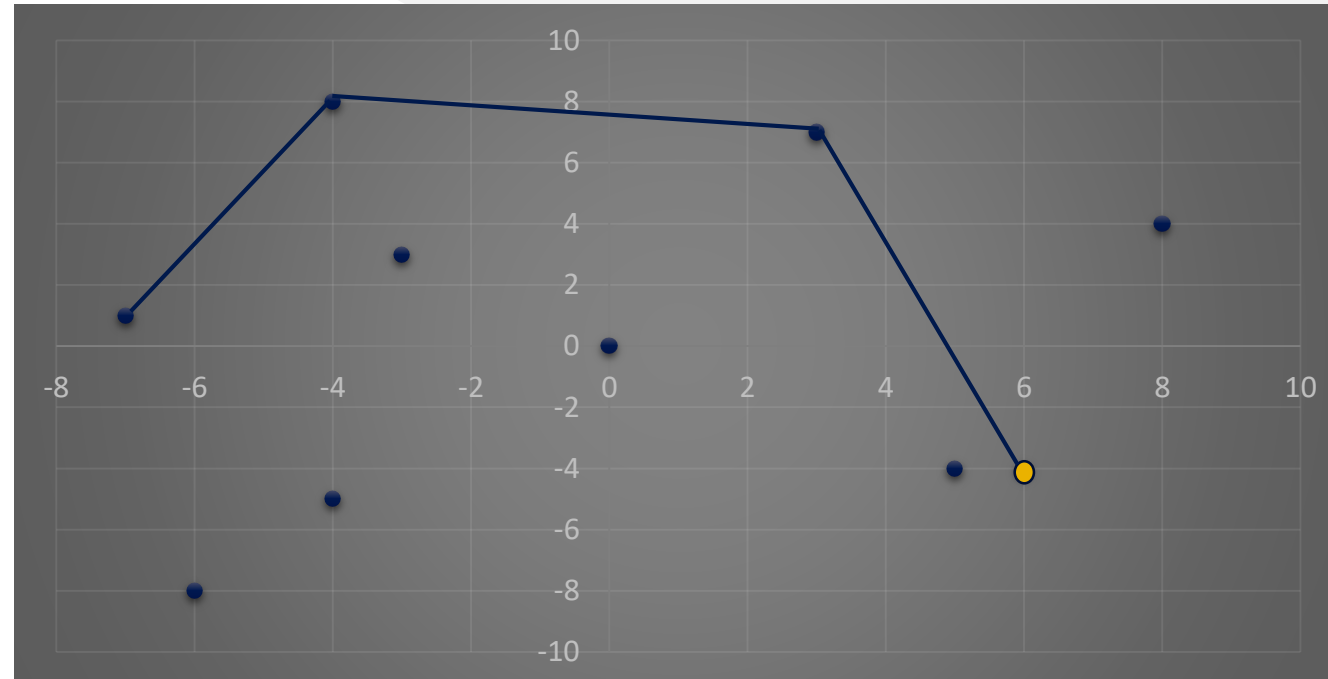


Coordinates: $(-7,1) \rightarrow (-6,-8) \rightarrow (-4,-5) \rightarrow (-4,8) \rightarrow (-3,3) \rightarrow (0,0) \rightarrow (3,7) \rightarrow (5,-4) \rightarrow (6,-4) \rightarrow (8,4)$

Vertices: $(-7,1) \rightarrow (-4,8) \rightarrow (3,7) \rightarrow (5,-4)$

Check: $p: (6,-4) \ q:(5,-4) \ r:(3,7) \ \rightarrow \ -22 > -33$

Algorithm

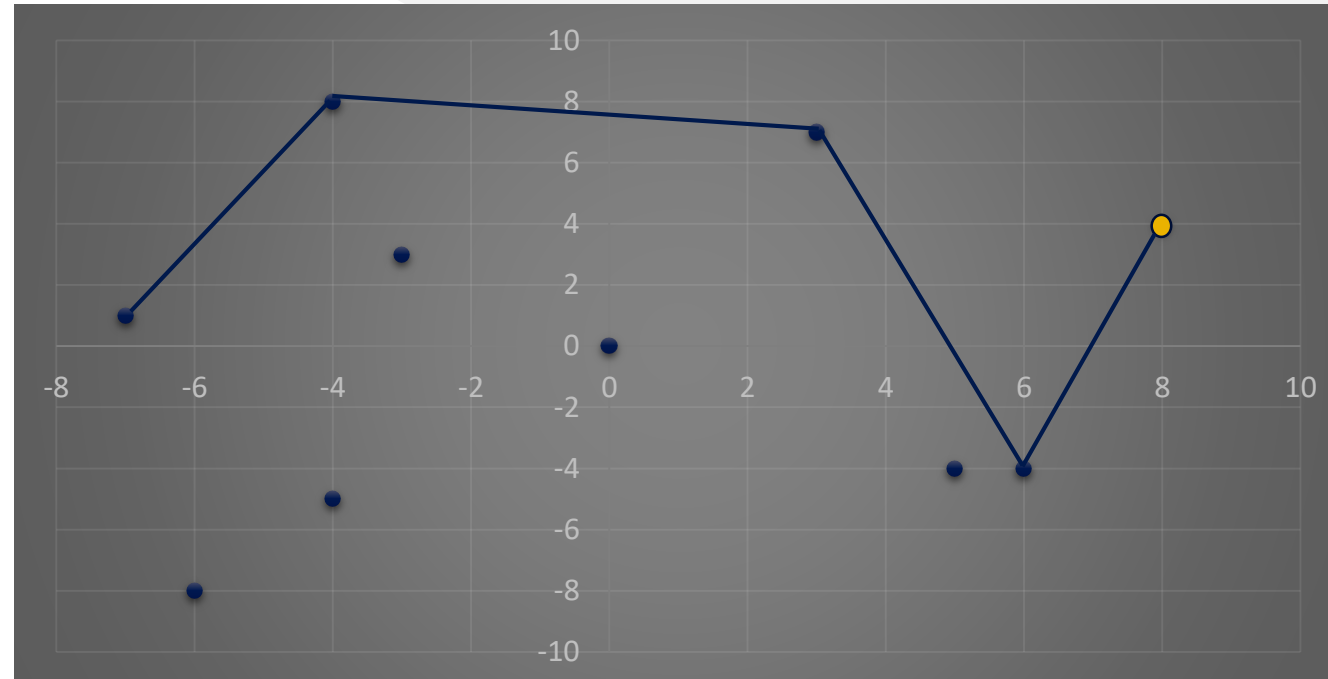


Coordinates: $(-7,1) \rightarrow (-6,-8) \rightarrow (-4,-5) \rightarrow (-4,8) \rightarrow (-3,3) \rightarrow (0,0) \rightarrow (3,7) \rightarrow (5,-4) \rightarrow (6,-4) \rightarrow (8,4)$

Vertices: $(-7,1) \rightarrow (-4,8) \rightarrow (3,7)$

Check: $p: (6,-4) \ q:(3,7) \ r:(-4,8) \ \rightarrow \ -120 < -10$

Algorithm

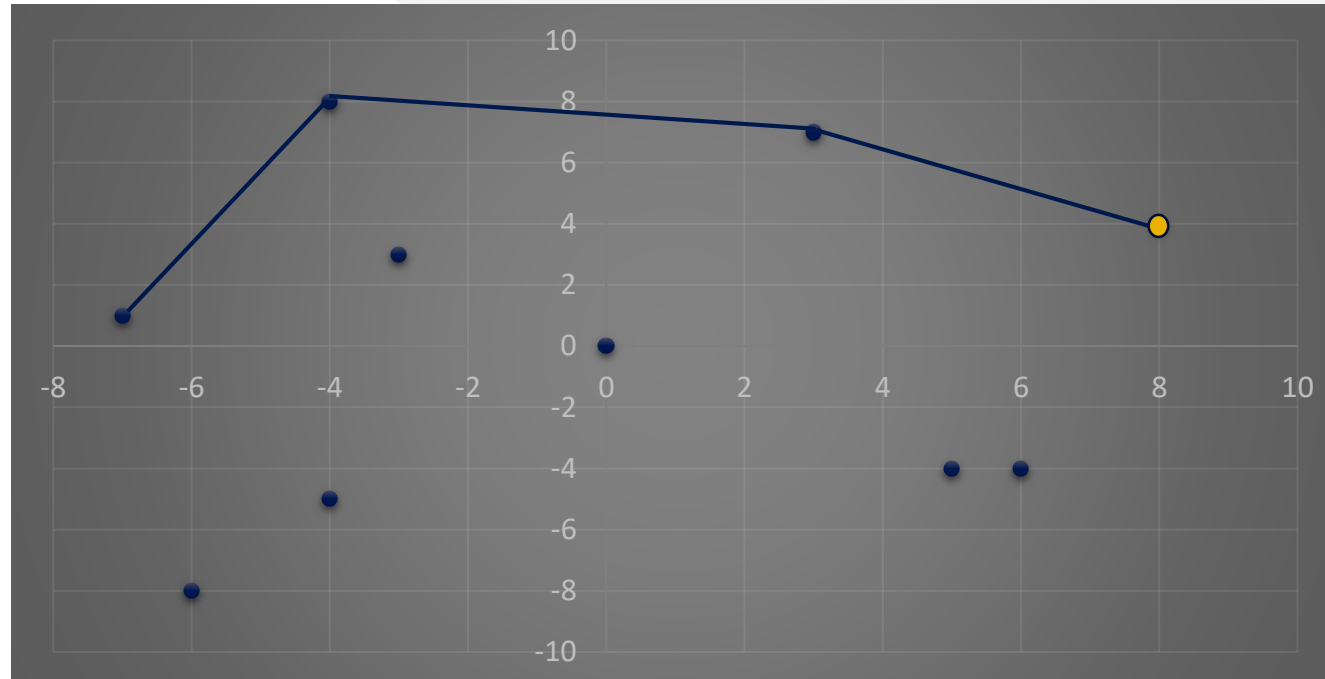


Coordinates: $(-7,1) \rightarrow (-6,-8) \rightarrow (-4,-5) \rightarrow (-4,8) \rightarrow (-3,3) \rightarrow (0,0) \rightarrow (3,7) \rightarrow (5,-4) \rightarrow (6,-4) \rightarrow (8,4)$

Vertices: $(-7,1) \rightarrow (-4,8) \rightarrow (3,7) \rightarrow (6,-4)$

Check: $p: (8,4) \ q: (6,-4) \ r: (3,7) \ \rightarrow \ -9 > -55$

Algorithm



Coordinates:



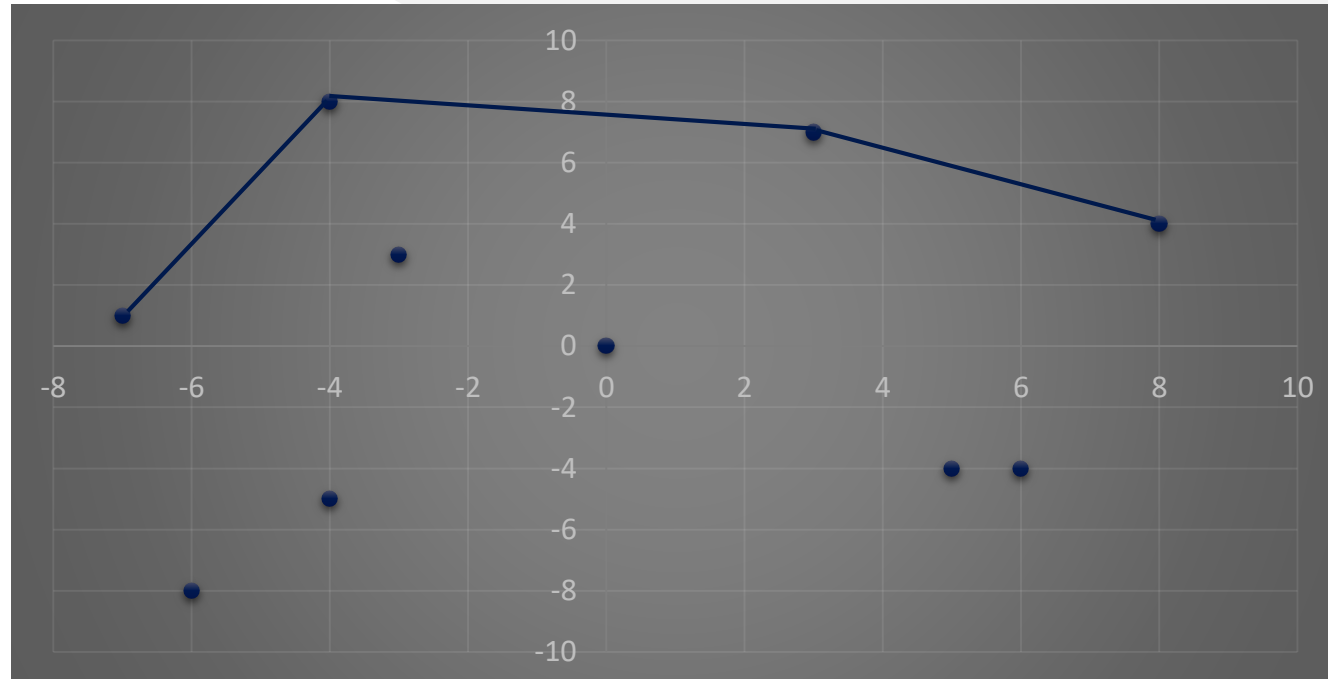
Vertices:



Check:

$p: (8,4) \quad q: (3,7) \quad r: (-4,8) \quad \Rightarrow \quad -28 < -12$

Algorithm



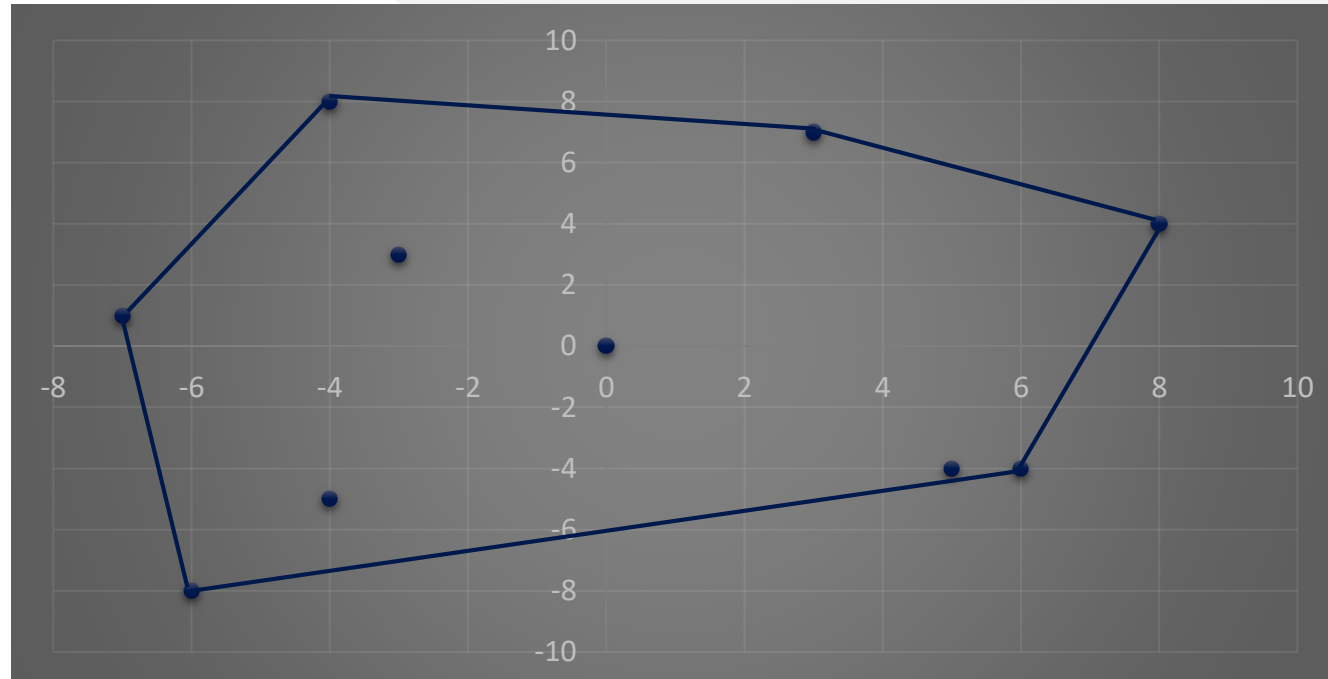
Coordinates:



Vertices:



Algorithm



Coordinates:



Vertices:





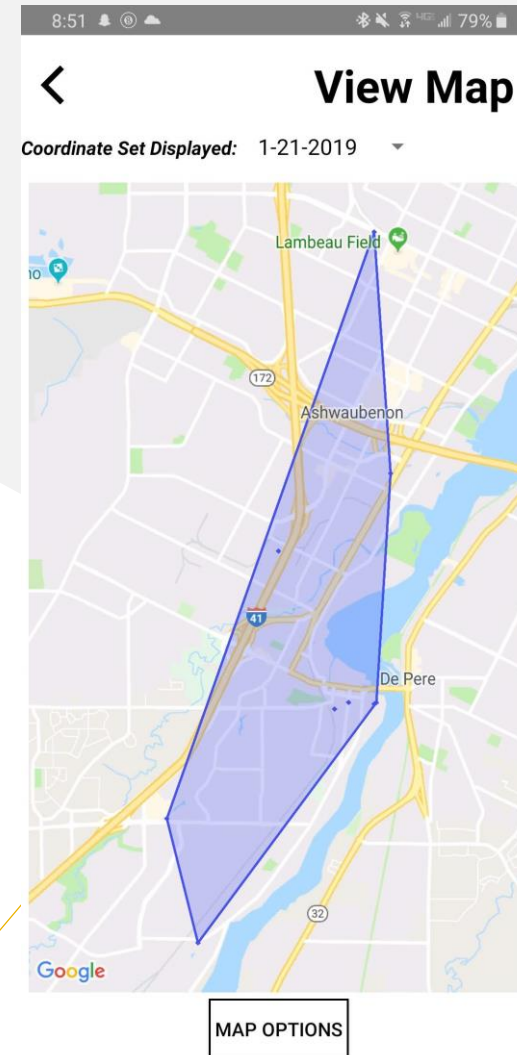
Capstone 2019

Part 3

Display the convex hulls as a map overlay

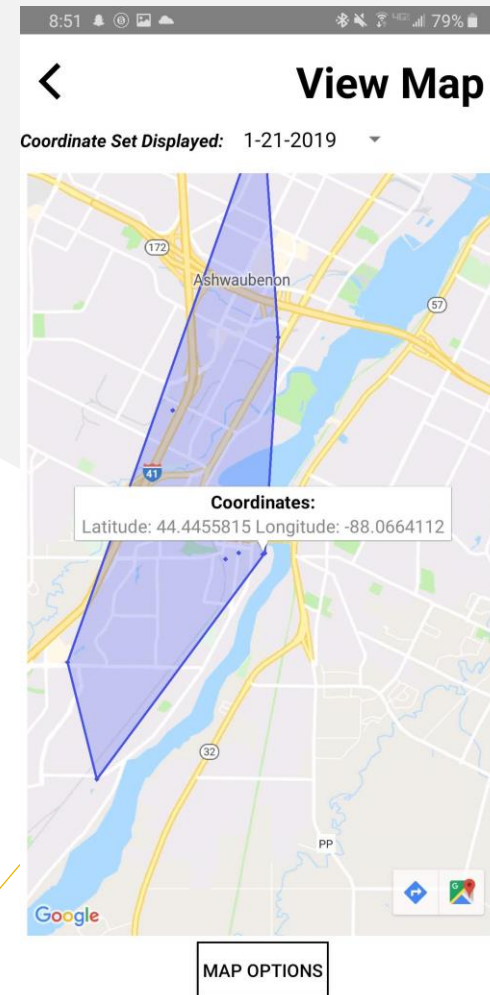
Map Overlay

- 3 different levels:
 - Google Maps *Polygon*
 - Google Maps *Groundoverlay*
 - Google Maps *Marker*
- Compute the center of convex hull.
 - Helps center the convex hull on the screen.
- Compute area of convex hull.



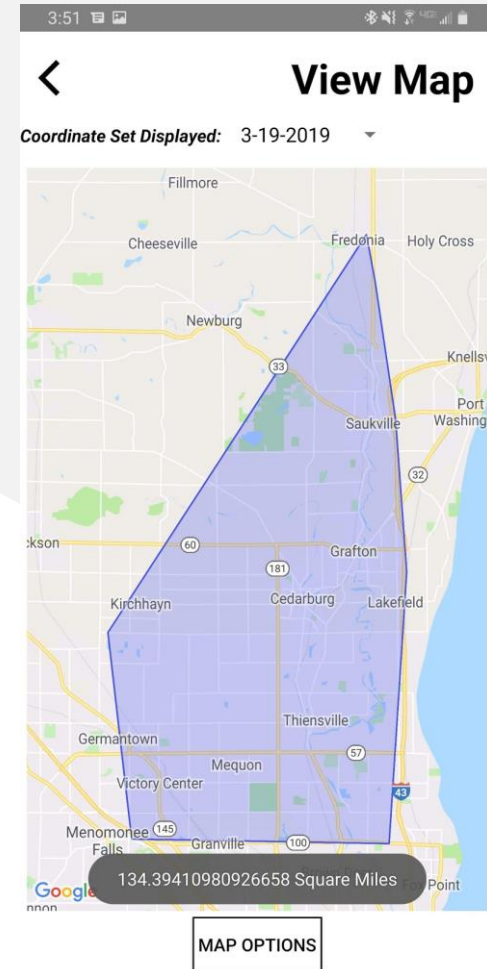
Map Overlay

- A project requirement is to display the coordinate of a vertex when requested.
- The app shows coordinate of any point in the convex hull when it is tapped
 - This is the purpose of the *Marker* objects in the convex hull



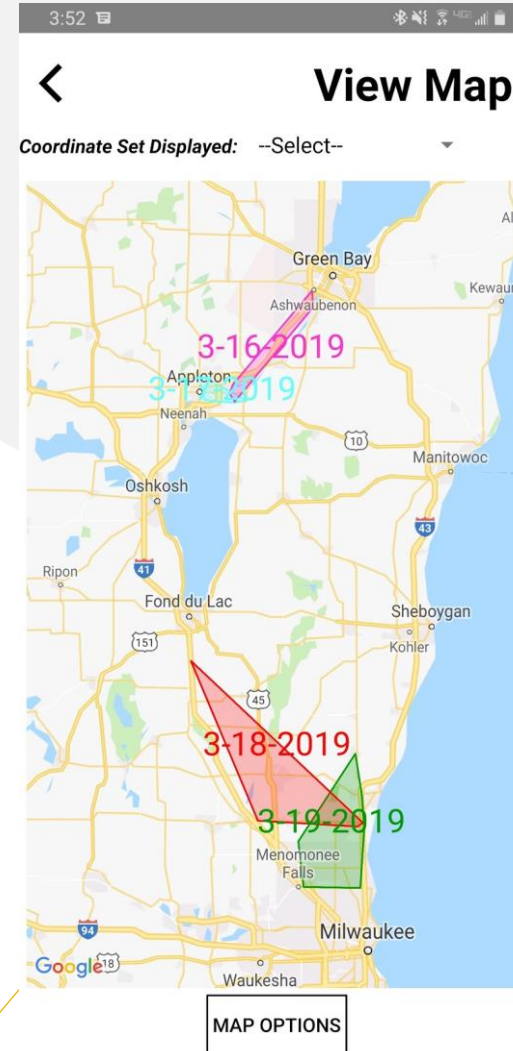
Map Overlay

- Displaying the area of a convex hull was another requirement.
- The app will display the area of a convex hull anytime a part of the convex hull that is not a coordinate is tapped.

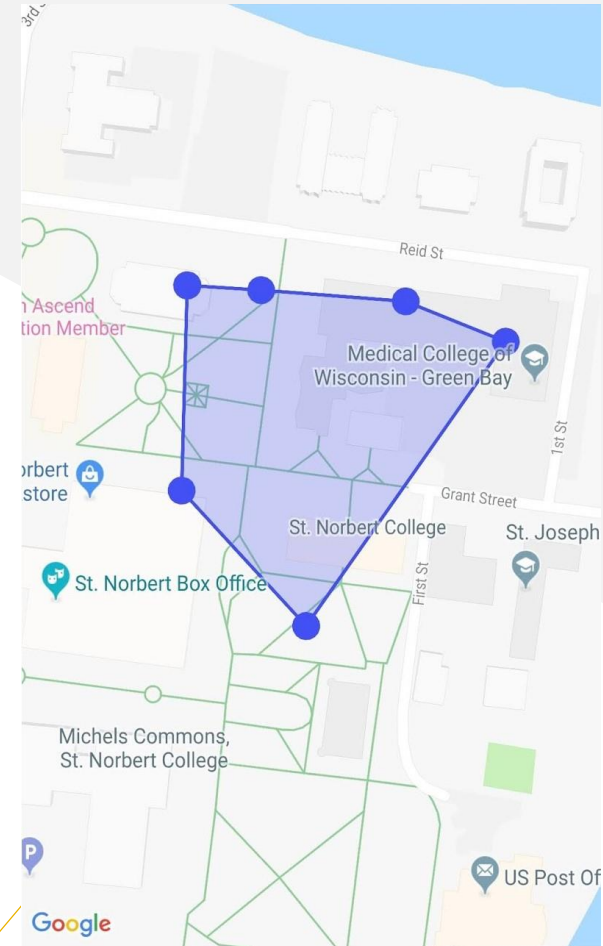
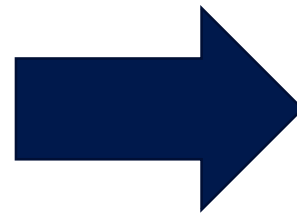
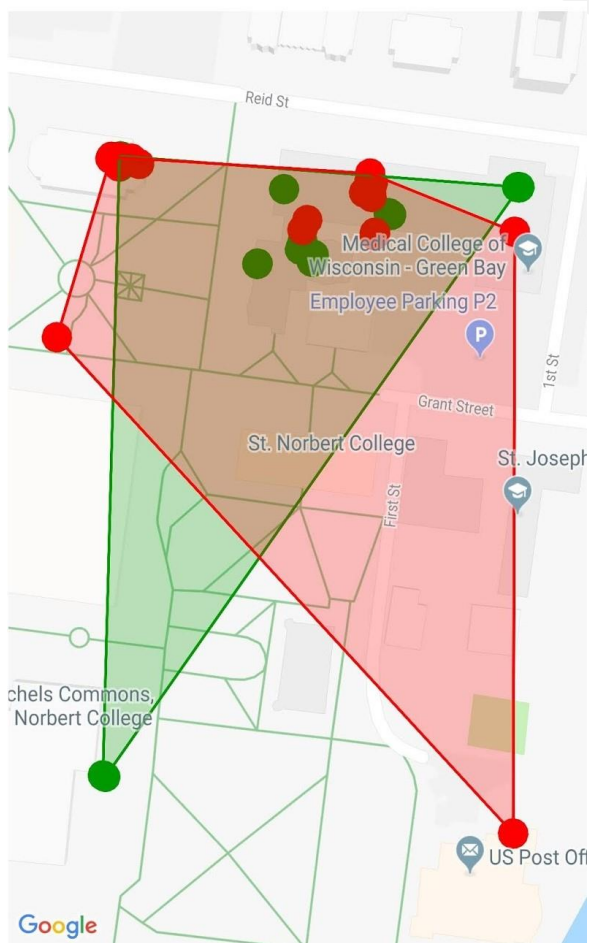


Map Overlay

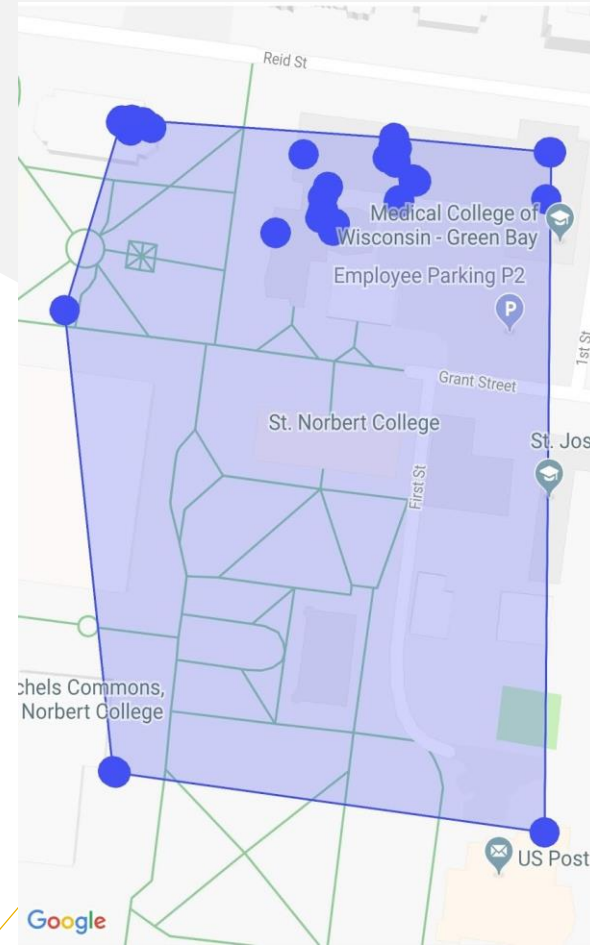
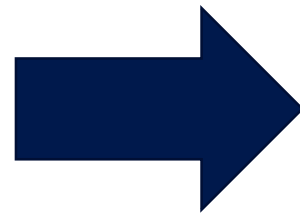
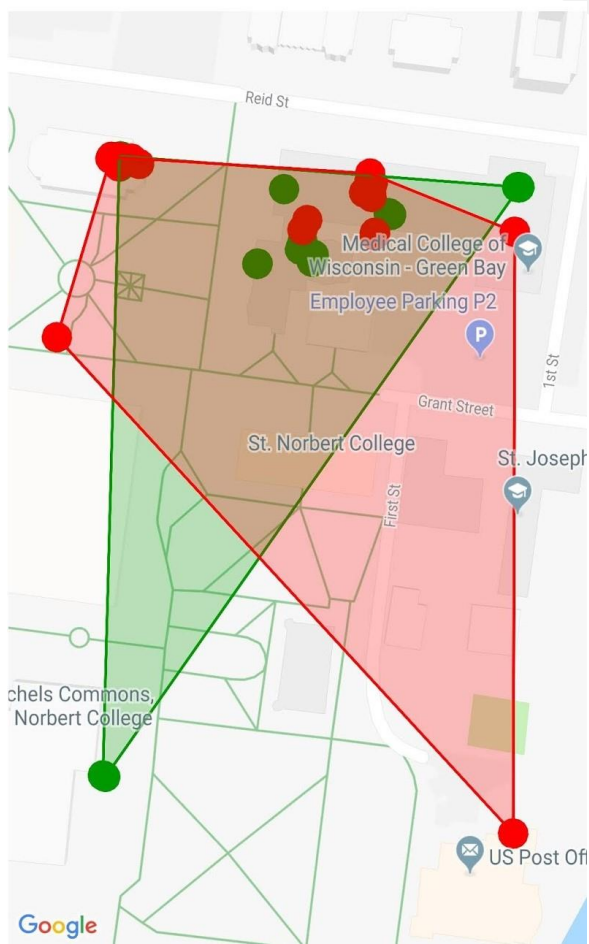
- Displaying each convex hull as a different color was another project requirement.
- When multiple convex hulls are displayed, each convex hull has a different color.
- Each convex hull will have a label as well.



Intersection



Union





Capstone 2019

Demonstration



Capstone 2019

Learning and Development

Learning

- Stack Overflow
- Android documentation is excellent for the most part when it comes to user interface.
- Google Maps documentation.

Extension

- Record the coordinates on a phone and write them to a database
- Use Google Maps API for web to construct the convex hulls
 - Google Maps API is much better supported on the web than on Android



Capstone 2019

Thank You!
Questions?