# Assignment 8: Solve a problem on your own.

## Learning Outcomes

* Solve a difficult problem on your own (I'll give you hints).

## Motivation

The Convex Hull problem is one that attempts to find the smallest convex area (or shape) that contains all points in a set. The "convex" part of convex hull means that the resulting shape doesn't curve inward at any point. There is a detailed wikipedia entry for [Convex Hull](https://en.wikipedia.org/wiki/Convex_hull) as well as a page of [Convex Hull Algorithms](https://en.wikipedia.org/wiki/Convex_hull_algorithms). The fastest algorithms solve this problem in .

Here's the general analogy of how to visualize the convex hull:

* You need a large board, a handful of nails (at least three), a hammer, and a large rubber band.
* Hammer all of the nails into one side of the board, but leave part of the nail head sticking out. Scatter the nails randomly.
* Take the rubber band and stretch it so that it wraps around all of the nail heads.
* The resulting shape of the rubber band is equal to the size and shape of the convex hull: it takes the smallest area that contains all of the nails and doesn't curve inward at any point. If you have exactly three nails, this will form a triangle. More nails may form other shapes.

## My Convex Hull Algorithm.

This approach is very simple and solves the problem in time.

* Start with a list of x,y locations, each representing one observation.
* Start with an empty list of edges. This will be returned as our convex hull.
* Repeat until there are no longer combinations of two points:
  + Pick out two points. Using some geometry, determine the line formed by these two points.
  + Store a variable called "above" and set it to 0.
  + Store a variable called "below" and set it to 0.
  + For every other point,
    - Project that point on the line formed by the first two points. In other words, determine the y-coordinate of the point if the x-coordinate were unchanged but the y-coordinate were moved to align with the line.
    - Compare the true y-coordinate of the point to the projection of the y-coordinate. If the true y-coordinate is less than the projected y-coordinate, then this point must be **below** the line. Increment below.
    - Else, this point must be on or above the line. Increment above.
  + In order for two points to be part of the edge of the convex hull, all other points must be above the line or all other points must be below the line. If so, add it to our list of edges. If there is a combination of above and below, then this is not an edge. Ignore it.
* Return the list of edges.

## Quick Geometry Lesson: Forming a line

A line is defined by a slope and a y-intercept and can be determined by using two points.

The slope of the line is formed by finding the difference in the rise over the difference of the run:

The intercept is formed by taking one of the two points and solving.

There is a problem that you will encounter if the slope is 0 (a purely vertical line). What should you do in this circumstance?

## Quick Geometry Lesson: Projecting a point onto a line

Suppose that you have a third point and you wish to project it onto a line. In a projection, you estimate the y-coordinate if the point was both on the line and existed at the x-coordinate:

In this case is the projected y-coordinate of the third point.

## Inputs

The input will begin with an integer representing the number of points. This number should be from 3 to 100. Call this n.

The next n lines will each contain two values representing (x,y) coordinates. It's best to assume that these values are floating point values (even if the samples are integers).

## Outputs

You should display all of the edges that make up the convex hull, one per line, represented by two points which are connected to form one edge.

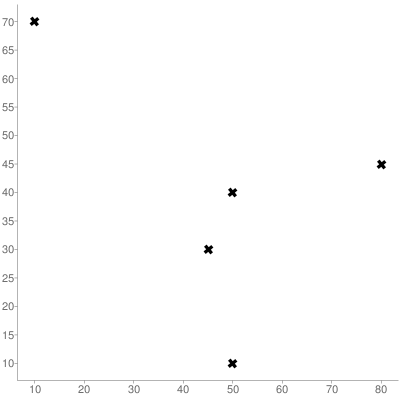
## Helpful tools

I use [Random.org](http://random.org/) in order to create 20 random points. I used [alcula.com](http://www.alcula.com/calculators/statistics/scatter-plot/) in order to generate the scatter plot images.

## Examples.

Example 1. Five Points

This is five points. There are three points on the convex hull. It should be a simple problem so solve.



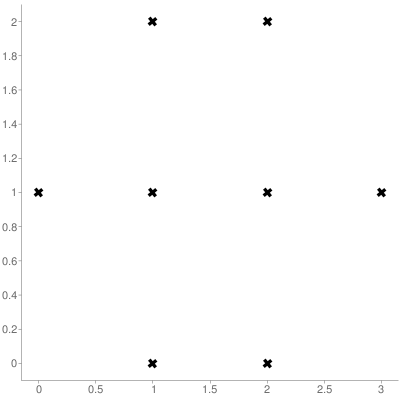
Five Points

5  
50 10  
80 45  
45 30  
10 70  
50 40

Results.  
(50,10) to (80,45)  
(50,10) to (10,70)  
(80,45) to (10,70)

Example 2. The Ruby

The ruby was created with horizontal and vertical lines which will potentially disrupt your algorithm.



Ruby

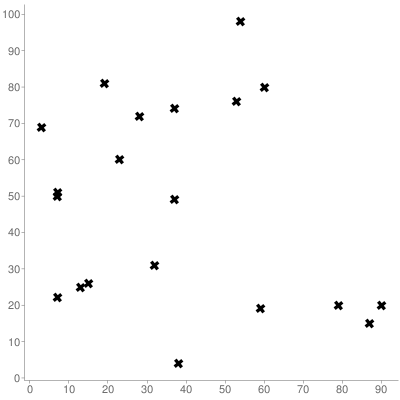
8  
0 1  
2 2  
1 0  
2 0  
1 2  
3 1  
1 1  
2 1

Results.

(0,1) to (1,0)  
(0,1) to (1,2)  
(2,2) to (1,2)  
(2,2) to (3,1)  
(1,0) to (2,0)  
(2,0) to (3,1)

Example 3. 20 Random Points

I went to Random.org and got 20 random points. You can do this too with as many points as you want.



Twenty Points

20  
59 19  
87 15  
7 22  
60 80  
53 76  
7 51  
3 69  
7 50  
23 60  
28 72  
90 20  
32 31  
13 25  
54 98  
15 26  
37 49  
38 4  
37 74  
19 81  
79 20

Results.

(87,15) to (90,20)  
(87,15) to (38,4)  
(7,22) to (3,69)  
(7,22) to (38,4)  
(3,69) to (19,81)  
(90,20) to (54,98)  
(54,98) to (19,81)

Example 4. One Thousand Sample Points.

I'm not going to post all those points here, but I will upload files containing my results. My program ran in 19.02 seconds. Try to beat the professor's time.

## Turn it in

Upload a zip file containing the files that use used or wrote yourself to the drop box on D2L:

* Make sure your name, CSCI 4270, and the Programming Assignment Number appear in comments in all of your files.
* Note: NO CREDIT will be given to programming assignments that do not compile.
* Make sure you have compiled and tested your program before handing it in.