

UMass Boston CS 240

Homework 8

Due 5/9/2019 17:00

1 Convex Hull

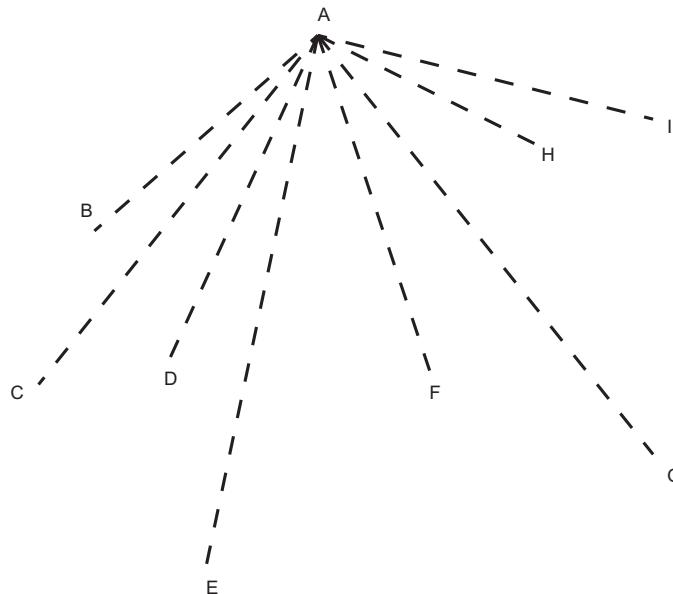
The assignment is to implement Graham's scan that computes the convex hull of a set of points in the two-dimensional space. The convex hull is the smallest convex set that contains the data set. For simplicity, we assume no three points in the data set are colinear.

1.1 Step 1: Find the First Point on the Hull

First, we find the point with the largest y-coordinate. It must be on the convex hull. Let us call it point A.

1.2 Step 2: Sort the Other Points

Second, we move the origin to point A by subtracting the x- and y-coordinates of A from the coordinates of other points. Then we sort all the other points by increasing polar angles, as shown in this figure.



You have probably learned the polar coordinate system in high school or in precalculus. If you need to refresh your knowledge, read https://en.wikipedia.org/wiki/Polar_coordinate_system.

When we put the origin at point A, the polar angles of all the other points are in the range of $[180, 360]$ degrees. Let point B be the one with the smallest polar angle. Point B must be on the convex hull.

1.3 Step 3: Graham's Scan

Graham's algorithm scans the other points in increasing polar angles: C, D, E, and so on. It starts by pushing points A and B into a stack:

A B

Consider the turn from \overrightarrow{AB} to \overrightarrow{BC} . We want to know whether this turn is a right turn or a left turn. In this case, it is a left turn – satisfying convexity – so we push C into the stack.

A B C

Next, consider the turn from \overrightarrow{BC} to \overrightarrow{CD} . It is a left turn, so we push D into the stack.

A B C D

Next, consider the turn from \overrightarrow{CD} to \overrightarrow{DE} . It is a *right* turn, violating convexity. So we pop D out of the stack, which becomes

A B C

Here we are still working on point E. So consider the turn from \overrightarrow{BC} to \overrightarrow{CE} . It is a left turn, so we push E into the stack.

A B C E

Graham's scan continues in this fashion, and the contents of the stack change as follows:

A B C E F

A B C E

A B C E G

A B C E G H

A B C E G

A B C E G I

At this point, we have scanned all points, and the convex hull is in the stack.

The remaining issue is to determine whether a turn is a left turn or a right turn. You learned it in high school, but you have probably forgotten. So here it is. The turn from \overrightarrow{AB} to \overrightarrow{BC} is determined by the cross product: $\overrightarrow{AB} \times \overrightarrow{AC}$. It is a left turn if the cross product is positive, and a right turn if negative. When the cross product is zero, the three points are colinear, which we assume does not happen. How do you calculate the cross product? Let the coordinates be $A = (A_x, A_y)$, $B = (B_x, B_y)$, and $C = (C_x, C_y)$.

$$\overrightarrow{AB} \times \overrightarrow{AC} = \det \begin{vmatrix} B_x - A_x & C_x - A_x \\ B_y - A_y & C_y - A_y \end{vmatrix} = (B_x - A_x)(C_y - A_y) - (C_x - A_x)(B_y - A_y).$$

1.4 Time Complexity

You will learn the formal definition of time complexity in CS 310. Let n be the number of points in the data set.

- Finding the first point takes $O(n)$ time.
- Sorting the other points takes $O(n \lg n)$ time.
- Scanning the points takes $O(n)$ time.

So the total time complexity is $O(n \lg n)$.

2 Implementation

Make a subdirectory `hw8` in your home directory, and copy all files from `/home/ming/240/hw8/`. The main program is the driver. You don't need to modify `main.c`, except for entering your information at the top of the file. The file `point.h` defines a point in the two-dimensional space. Each point has its x - and y -coordinates. The points are stored in an array. You may add new definitions to `point.h`.

Write your code in `graham.c`. Before returning to `main()`, the function `graham()` should print the number of vertices on the hull and list the vertices in counterclockwise order, starting from the vertex with the largest y -coordinate, in the following format:

```
Out of 10 points, 3 vertices are on the convex hull
A (-0.1234,  0.9876)
B (-0.6789, -0.4567)
C ( 0.5678,  0.1234)
```

Write a `Makefile` to compile your source code to the executable `graham`. Make sure your code follows the style guidelines. Write a `readMe.txt` that discuss what you found difficult about this assignment and what you learned completing it.

3 Grading Rubric

1. (10 points)
 - (a) Existence of the directory `/home/user??/hw8`
 - (b) Inside `hw8`: `point.h`, `main.c`, `graham.c`, `Makefile`, `readMe.txt`
 - (c) Other files are allowed
2. (30 points) Coding style in `*.c` and `*.h`
 - (a) (10 points) Adherence to Style Guidelines, esp. header comments, function comments
 - (b) (15 points) Sensible pseudocode for `graham.c`
 - (c) (5 points) `Makefile` adheres to the `Makefile` style as given and all commands work
3. (40 points) Correctness
 - (a) (10 points) `./graham`, `./graham -n 9`, and `./graham -n 100` run without error
 - (b) (5 points) Code sets vertex with highest y value as letter A positioned at (0,0)
 - (c) (10 points) Code outputs correct number of vertices on the convex hull
 - (d) (10 points) Code outputs correct CCW order of vertices identified by letter
 - (e) (5 points) No typos or awkwardness in output presentation
4. (20 points) `readMe.txt`
 - (a) (10 points) Describe several difficulties encountered.
 - (b) (10 points) What have you learned from each?