Topcoder SRM 641, D1, 250-Pointer "TrianglesContainOrigin"

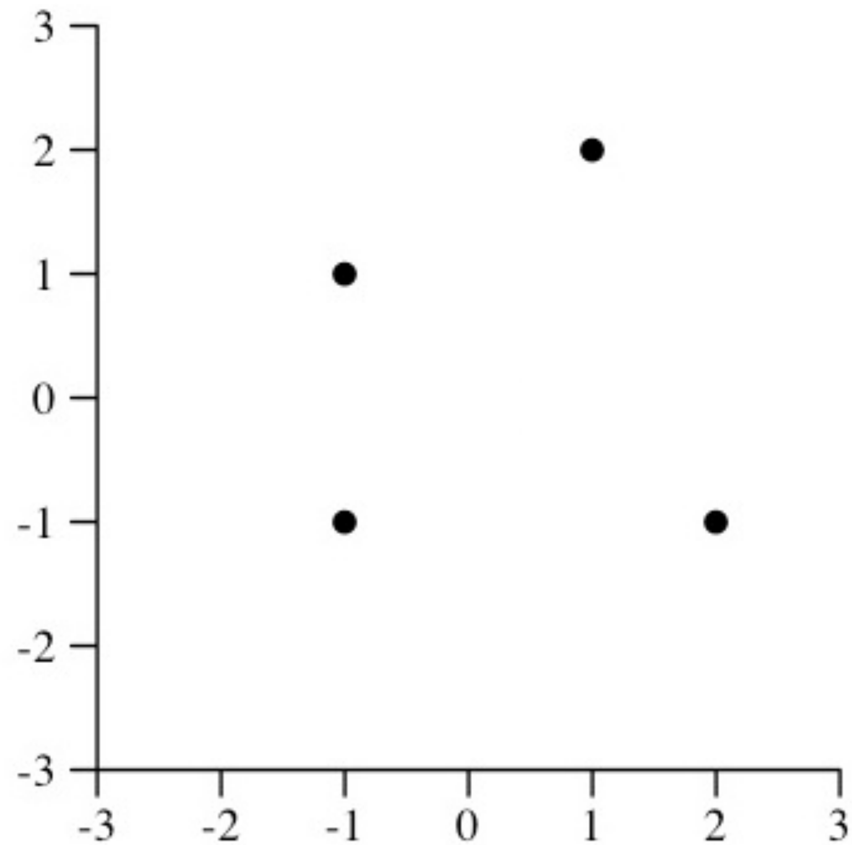
James S. Plank
EECS Department
University of Tennessee

CS494 Class
September 3, 2020

The problem

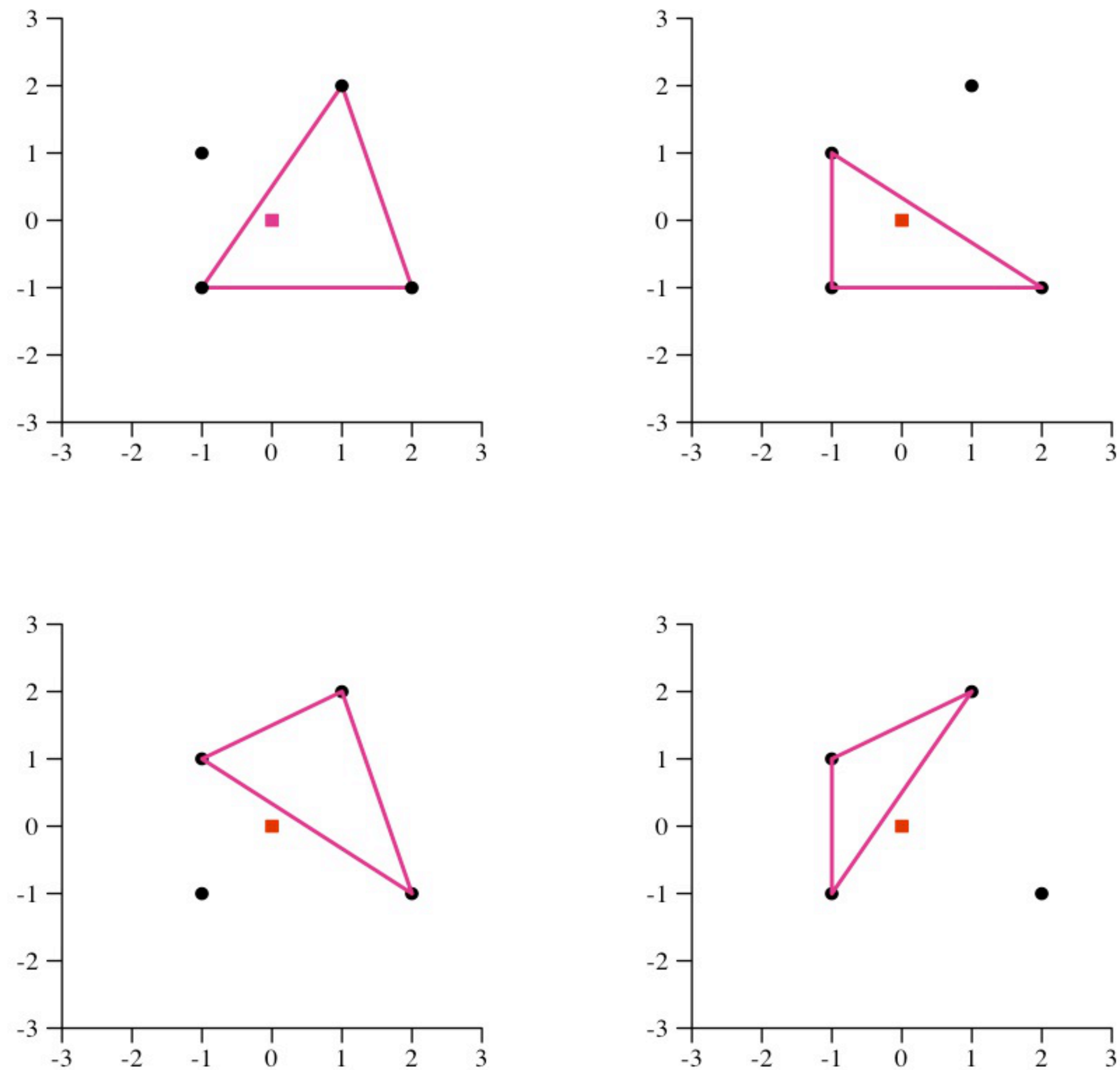
- You are given the (x,y) values of points on a two-dimensional grid:

<u>Example 1:</u>
(-1,-1)
(-1, 1)
(1, 2)
(2, -1)



The problem

- Of all the triangles with these points as endpoints, how many have the origin inside?



- <u>Example 1:</u> There are 4 triangles, two of which include the origin.

Prototype and Constraints

- **Class name**: `TrianglesContainOrigin`
- **Method**: `count()`
- **Parameters**:

x	`vector <int>`	X coordinates
y	`vector <int>`	Y coordinates

- **Return Value**: `long long`
- **Constraints**:
  - `x.size() == y.size()` ≤ 2500.
  - `x` and `y` values between -10,000 and 10,000.
  - No three values co-linear
  - No two values co-linear with the origin.

Brain-dead enumeration of triangles

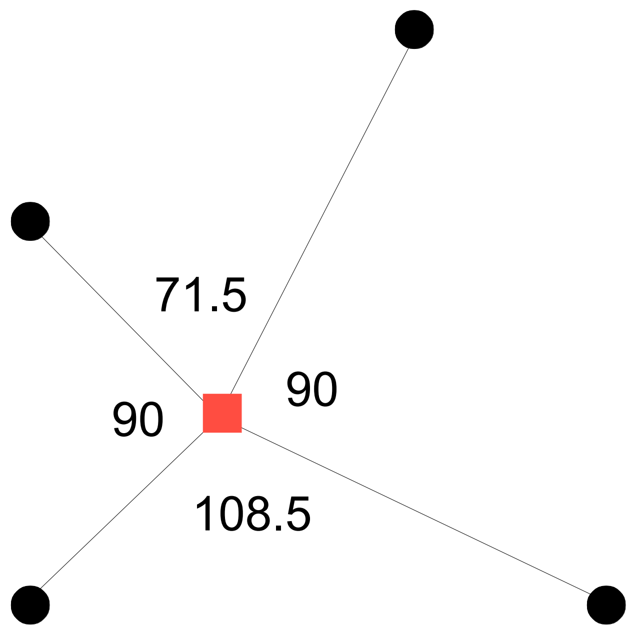
- Let n = `x.size()`.
- Then the number of triangles is:

$$\binom{n}{3} = O(n^3)$$

- When n = 2500, this is 2,590,630,000. Too slow.
- Our solution can be $O(n^2)$, but not much slower.

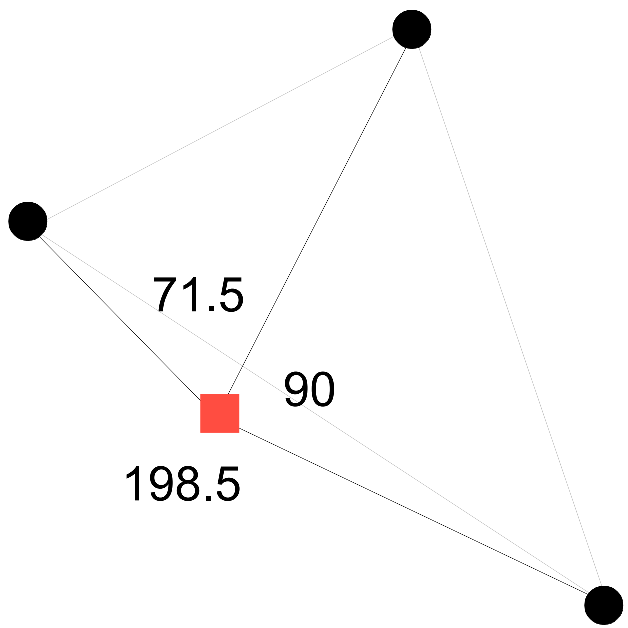
The Key Insight

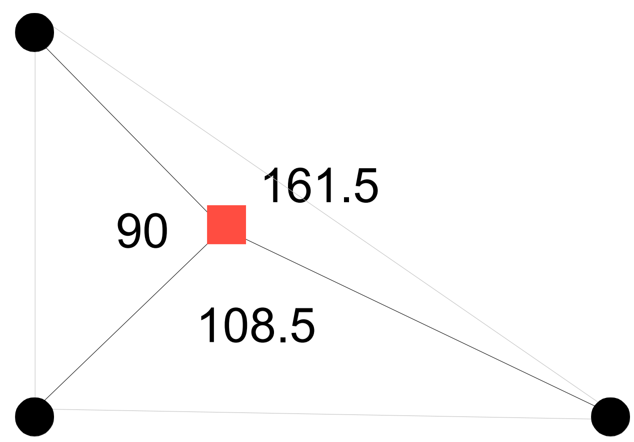
- Draw a line from the origin to each point, and then calculate the angles of adjacent lines:



The Key Insight

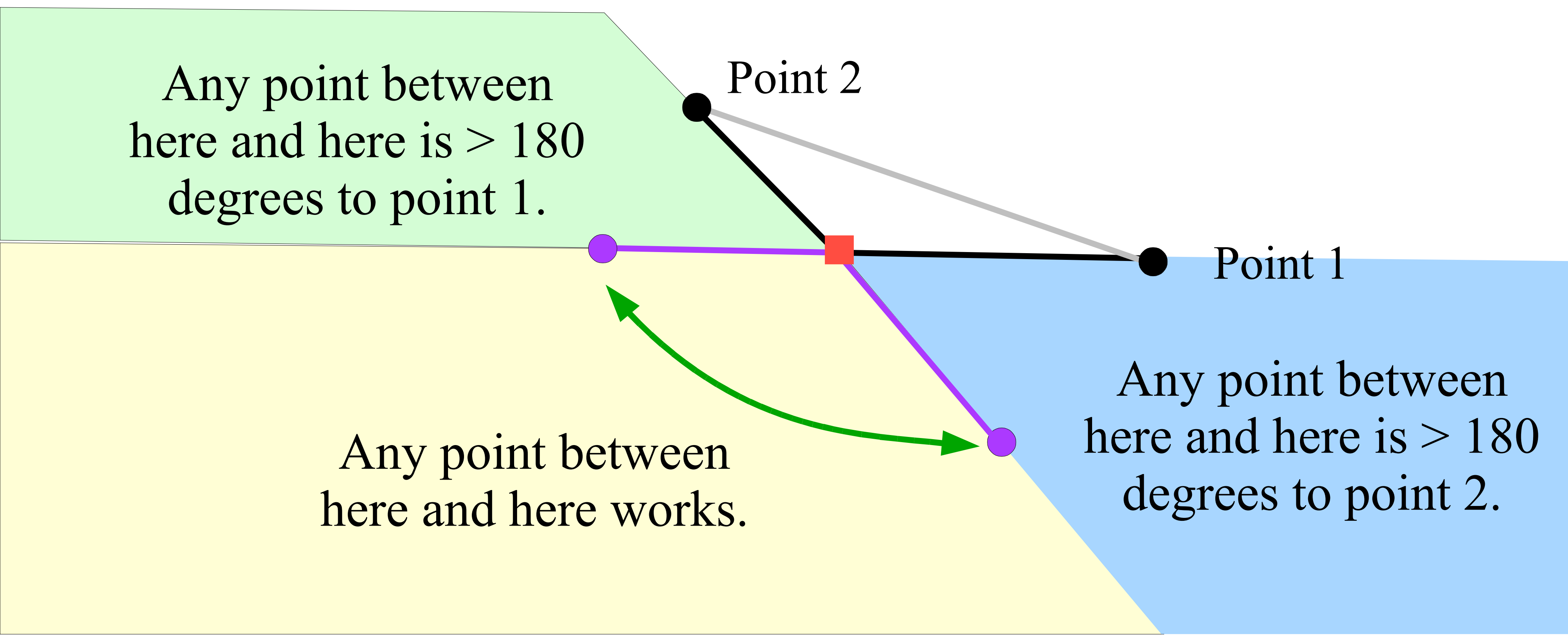
- Consider a triangle – it will include the origin if and only if each of the angles of lines from the origin is less than 180:





The Strategy

- Enumerate all pairs of points whose angle to the origin is less than 180 degrees:
- For a given pair, there is a minimum and maximum angle that the third point can have.

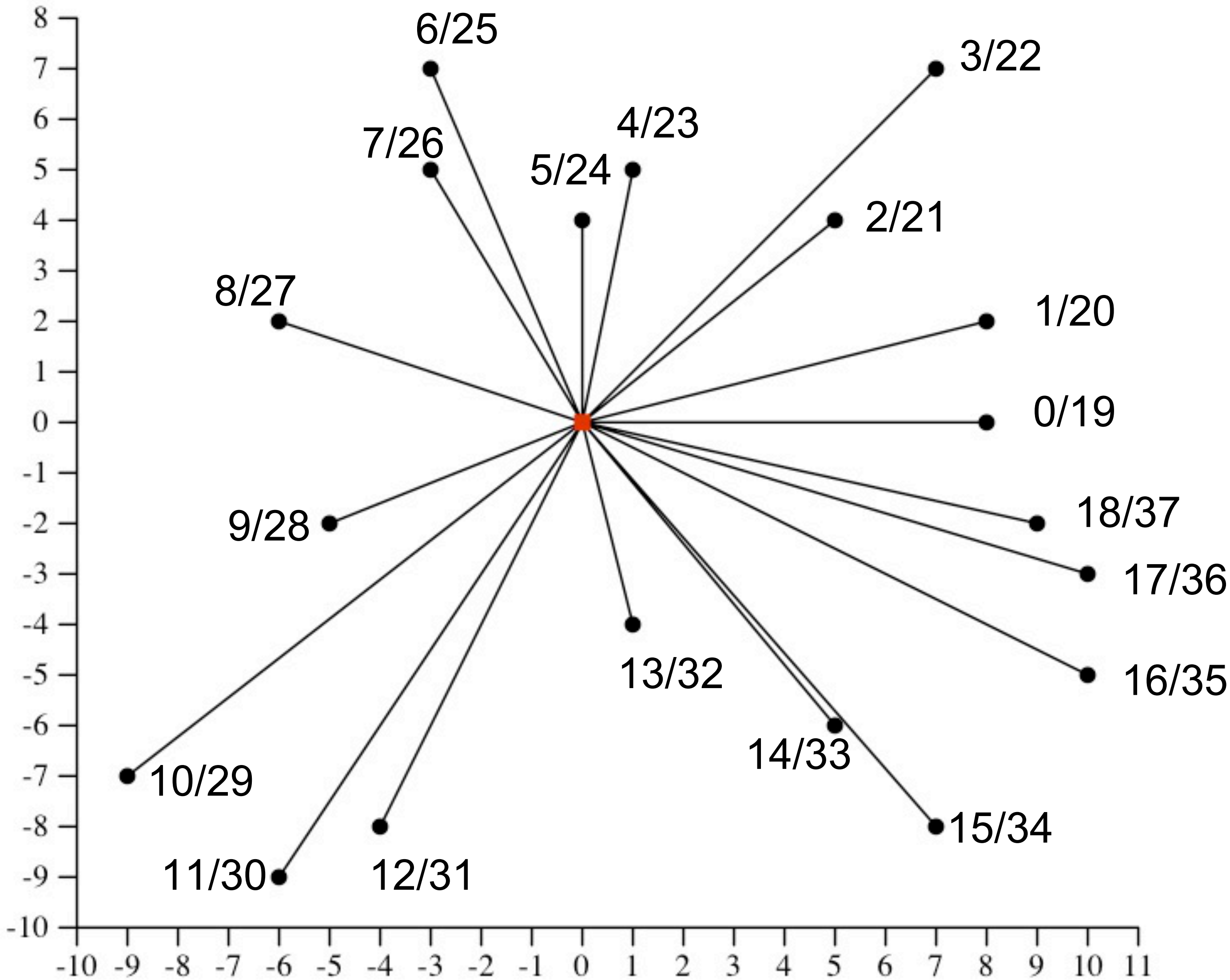


An Algorithm

- For each point, calculate the point's angle *a* from the origin.

- Insert the points into a map keyed by angle.
- Also insert the points keyed by angle+360.
- Number the points in the map by ascending angle.

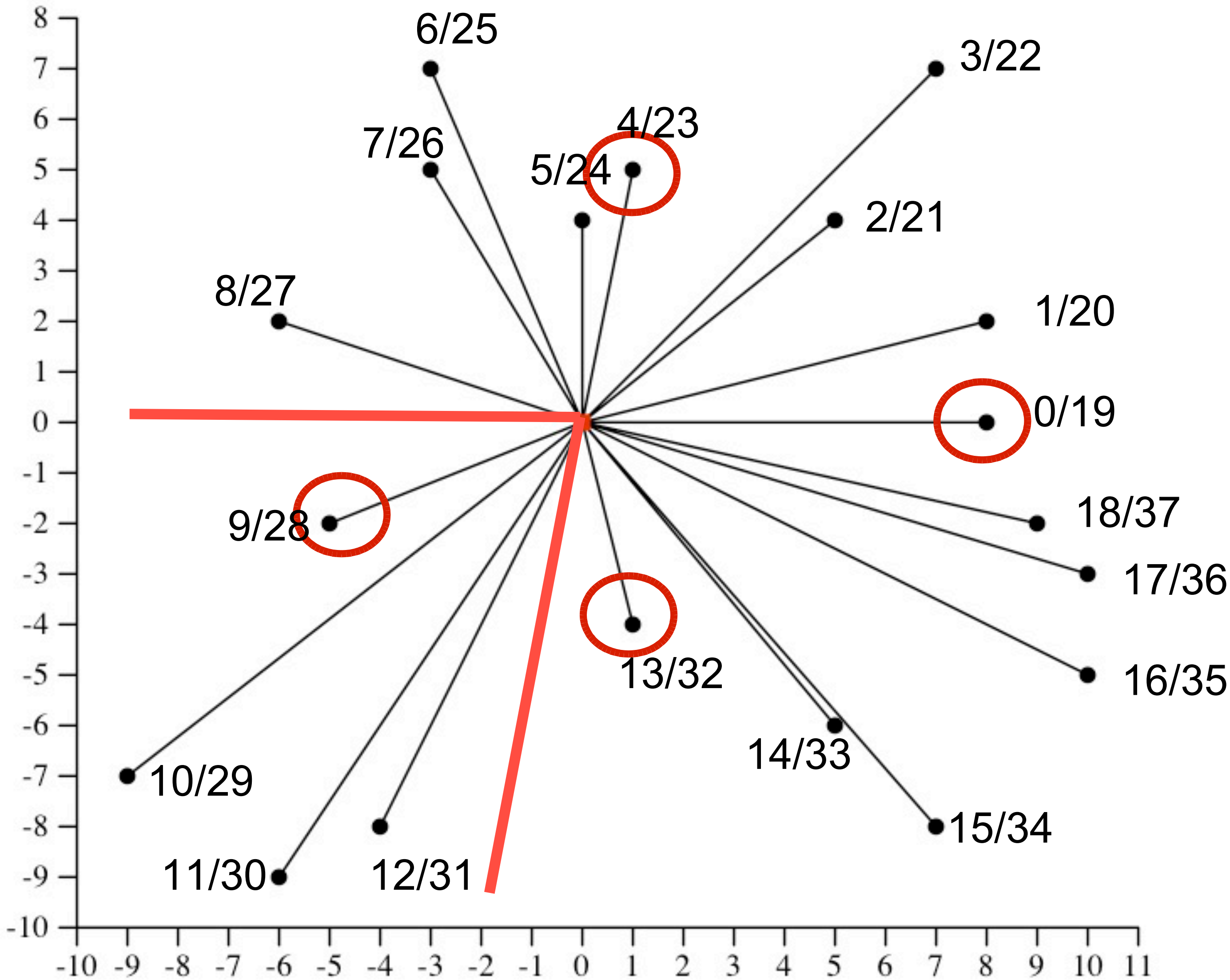
Let's look at example 3



An Algorithm, Continued

- For each point x < 360 and each point y whose angle to x is less than 180 degrees, use *upper_bound()* to find:
    - The smallest point whose angle is > 180 to x.
    - The smallest point whose angle is > 180 to y.
    - The difference in vals is the number of points that can complete the triangle!
- Sum the triangles and divide by 3 for the answer!

Let's look at example 3



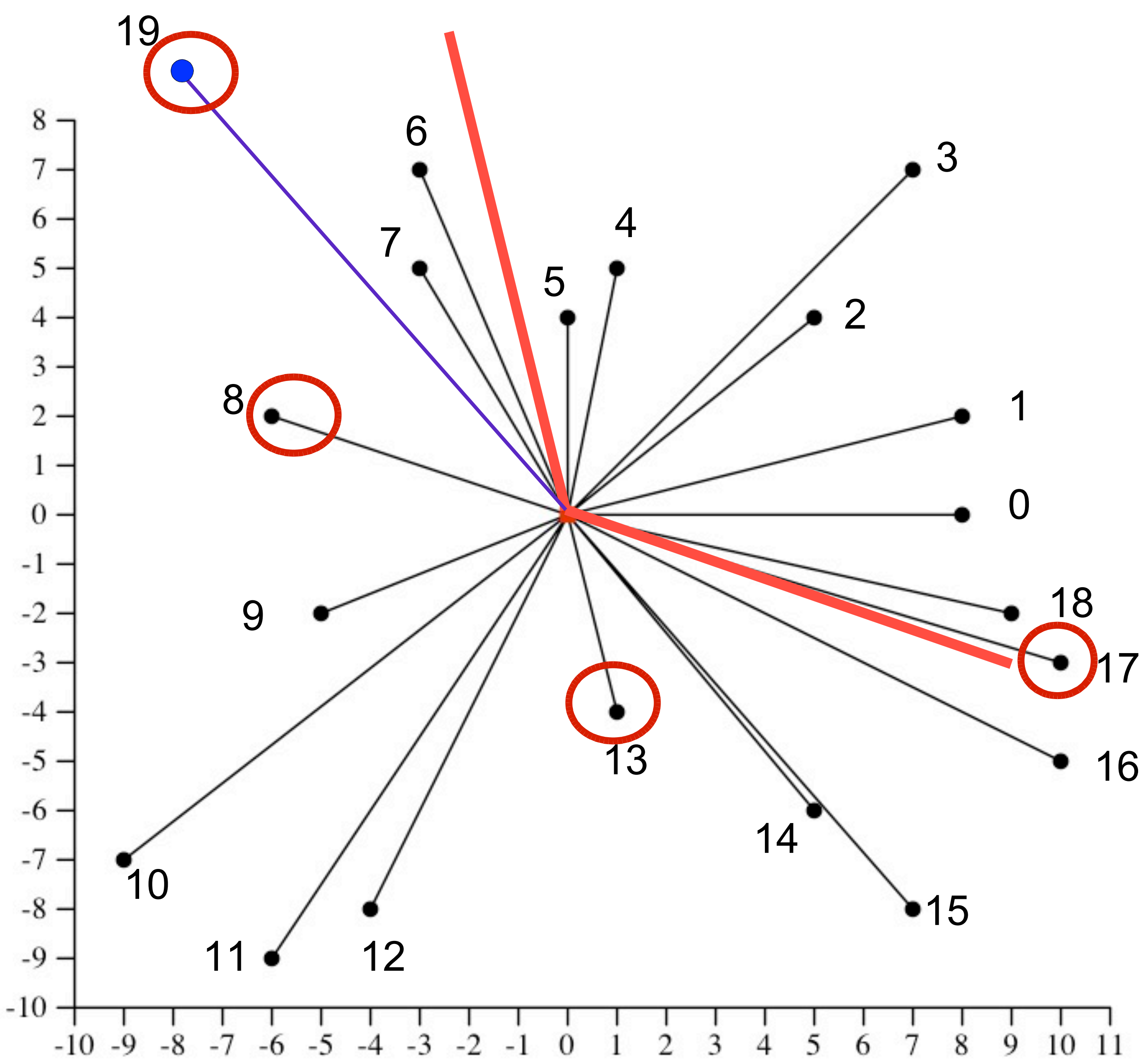
Consider points 0 and 4.

Upper-bound will find points 9 and 13.

Therefore, there are four points that can complete the triangle with points 0 and 4.

Improvements

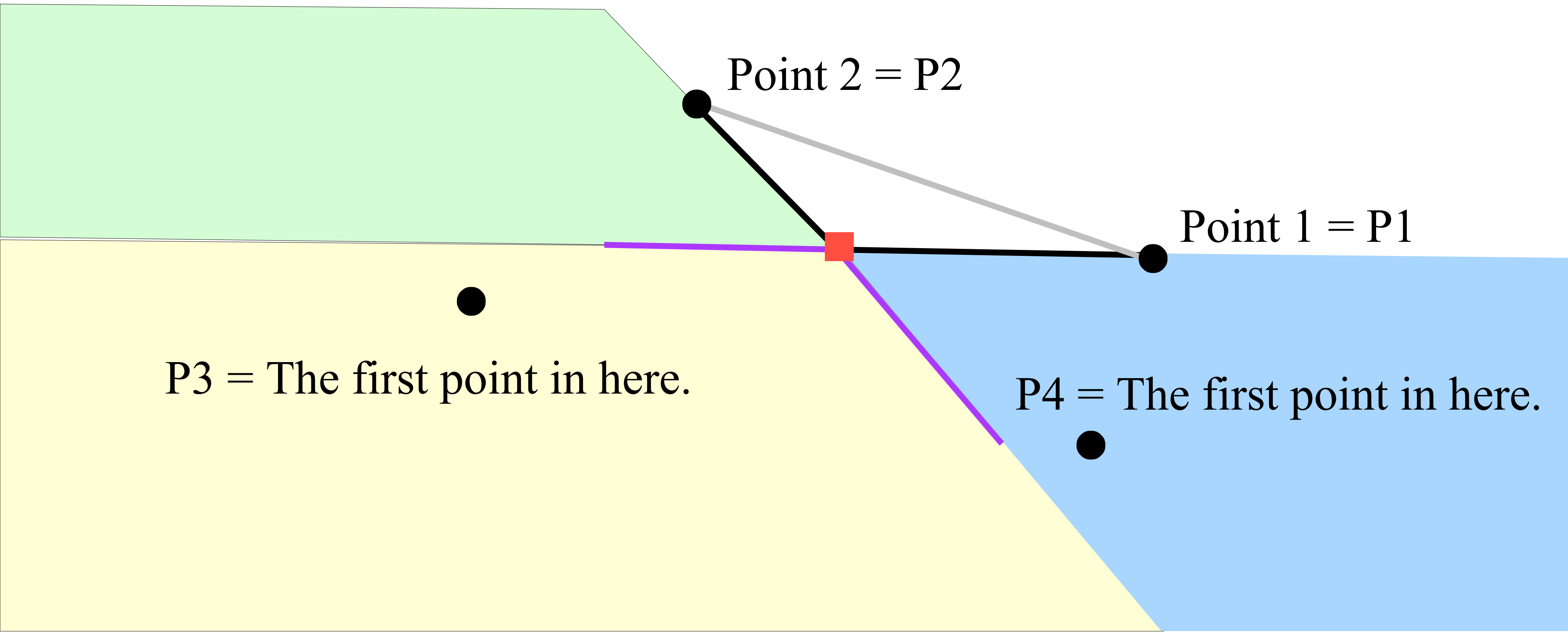
- Insert a giant sentinel, and:
- You only have to insert each point once.
- You don't have to divide by three.
- (0 triangles when b-a > 180)



Time for some clicker questions

We can make it faster.

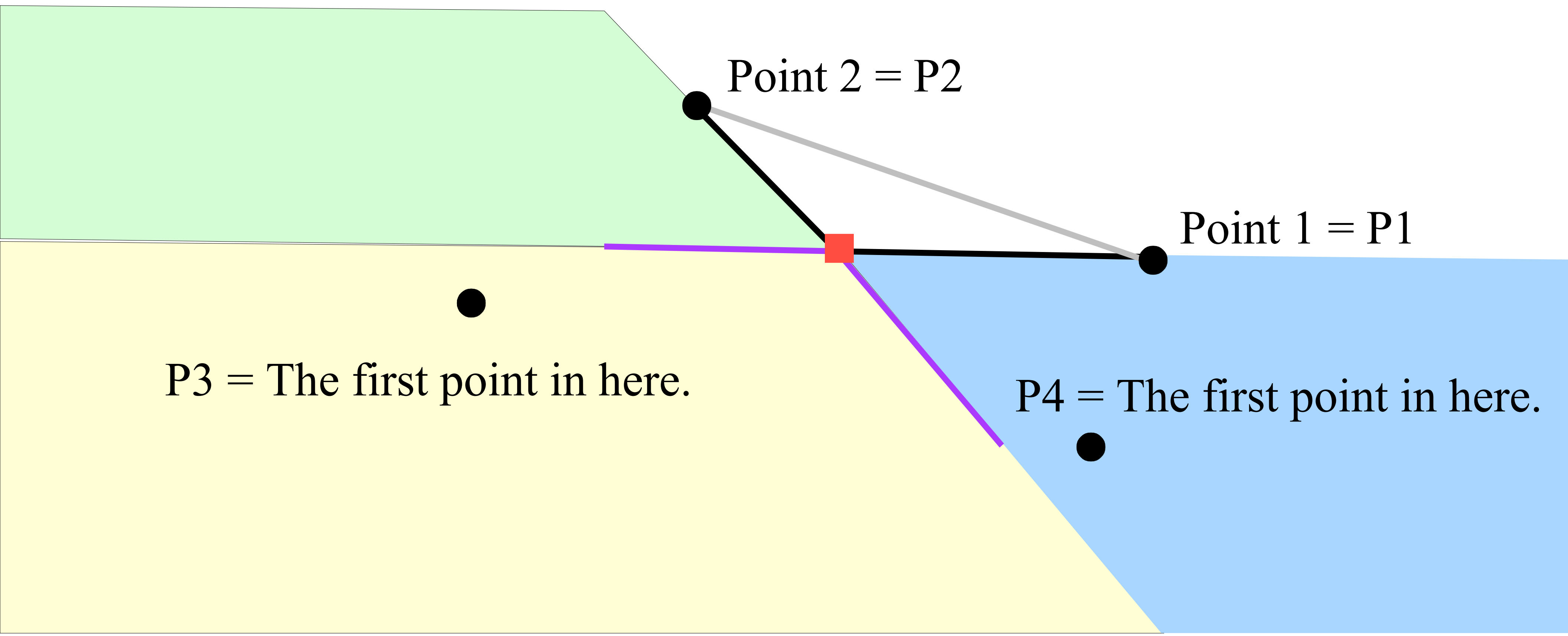
Label the points: P1, P2, P3, P4



We can make it faster.

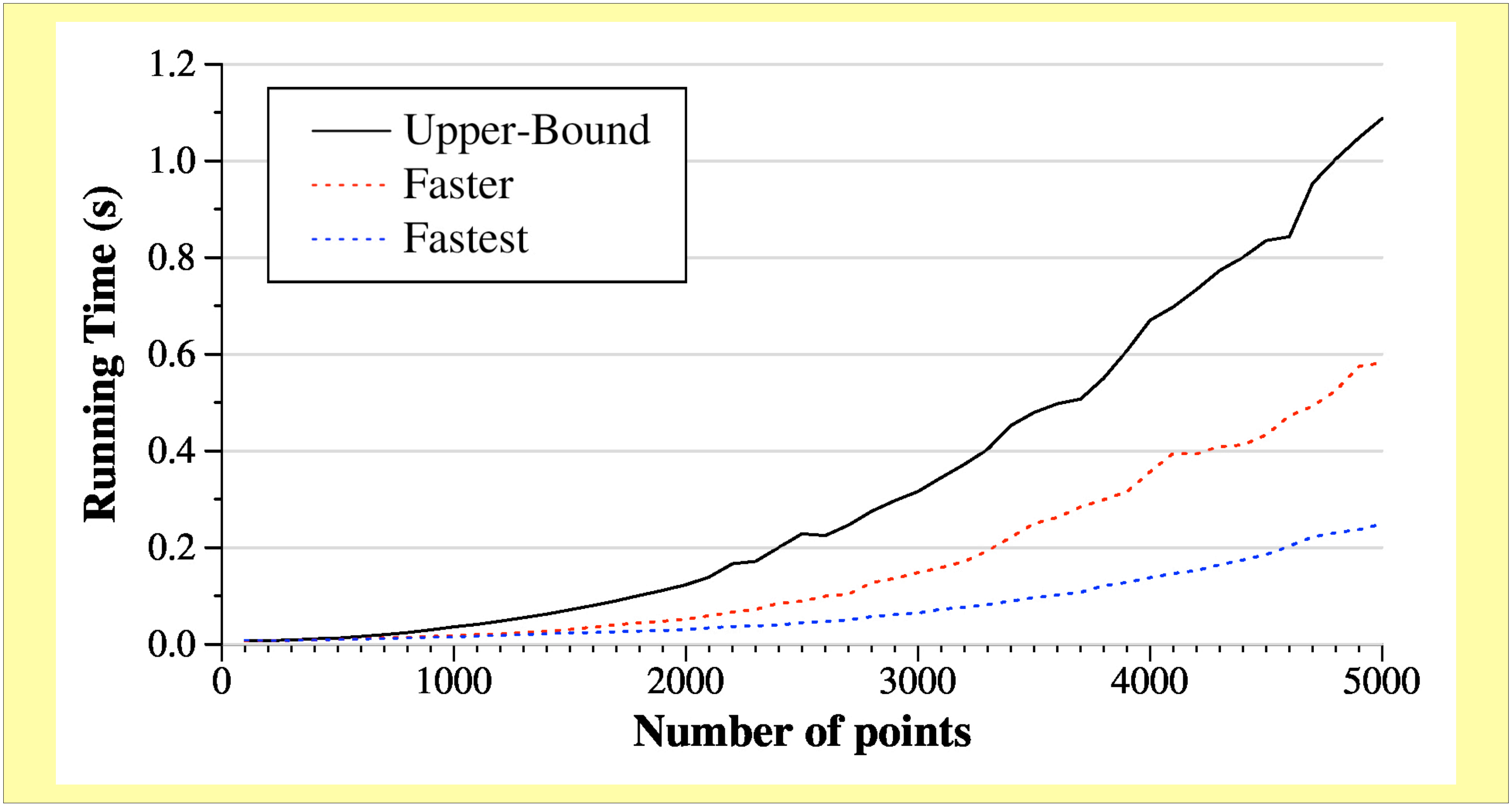
Label the points: P1, P2, P3, P4.

- The number of triangles for (P1,P2) is `(P4->val - P3->val)`
- Set P3, and set P4 = P3 when you set P1.
- For each P2, increment P4 until it's right.
- A total of $O(n)$ increments for all P2 makes it $O(n^2)$ overall.



Experiment

- MacBook Pro 2.2 Ghz, optimized with -O2
- Increments of 100, each point the average of ten runs.



How did the Topcoders Do?

- 580 competitors
- 285 (48%) submitted a solution.
- 216 (76%) of the submissions were correct.
- That's 38% - I suspected this one would be hard!

(For class)

- Go over the program that makes the jgraph (in the lecture notes directory).
- But delete the part that does the calculation, so you can do the calculation live (maybe even give the students a chance to do it).
- Remember txt/points-500.txt and txt/points-2500.txt.

Topcoder SRM 641, D1, 250-Pointer "TrianglesContainOrigin"

James S. Plank
EECS Department
University of Tennessee

CS494 Class
September 3, 2020