# Geometry for the Chop Problem

Here we describe the geometric primitives needed to do this problem.

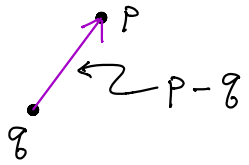
## 1 Operations on points and vectors

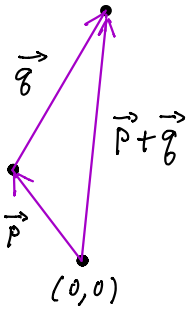
In computational geometry, the most primitive object is a *point*, which we can represent as a tuple of floating point numbers (or integers) In 2D geometry points are pairs $(x, y)$. A vector is also a pair of numbers, which is the same representation as a pont. Indeed, the difference between a point and a vector is often just context, points are typically used to denote locations, and vectors are typically used to denote directions/lengths or differences, but mathematically they are interchangeable.

**Addition and subtraction** Let's start by defining pointwise addition/subtraction of vectors:

$$(x_1, y_1) + (x_2, y_2) \coloneqq (x_1 + x_2, y_1 + y_2)$$
$$(x_1, y_1) - (x_2, y_2) \coloneqq (x_1 - x_2, y_1 - y_2)$$

The effect of vector addition and subtraction can be visualized as in the following pictures. $p - q$ results in a vector that points from the point $q$ to the point $p$. Adding $p + q$ results in a vector that points to the location you would get to if you start at $(0,0)$ then follow $p$ then $q$ (equivalently, $q$ then $p$).



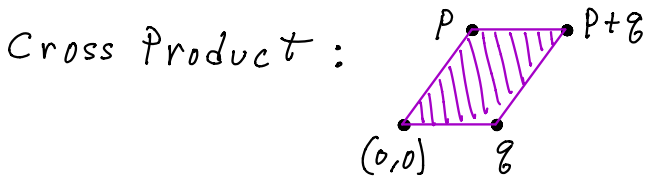




**The cross product** The cross product is a kind of "multiplication" of two points. In 2D, the cross product is defined as

$$(x_1, y_1) \times (x_2, y_2) \coloneqq x_1 y_2 - y_1 x_2$$

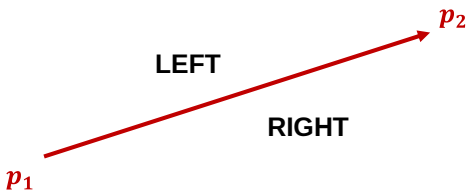
The cross product of $p$ and $q$ is the signed area of the parallogram with the four vertices $(0,0), p, q, p+q$. Equivalently, it is twice the signed area of the triangle with vertices $(0,0), p, q$. The sign is determined by the "right hand" rule. This means that if the angle at $(0,0)$ starting at $p$, going

clockwise around $(0,0)$ and ending at $q$ is less than 180 degrees, the cross product is negative, otherwise it's positive. It's zero of the angle between $p$ and $q$ is 0 or 180 degrees.

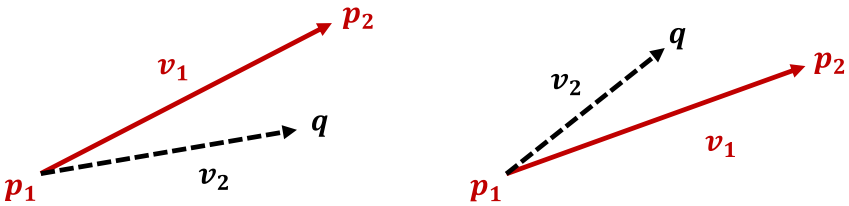



## 2 Line-side test

The line-side test is the next most fundamental primitive we will introduce, and turns out to be a critical ingredient in a lot of geometry algorithms. Given three points $p_1, p_2, q$, the output of the *line side test* is 1 (or LEFT) if the point $q$ is to the left of ray $p_1 \rightarrow p_2$, $-1$ (or RIGHT) if the point $q$ is to the right of ray $p_1 \rightarrow p_2$, and 0 (or ON) if it is on that ray.




The algorithm is to construct vectors $v_1 = p_2 - p_1$ and $v_2 = q - p_1$, then take the cross product of $v_1$ and $v_2$ and look at its value compared to 0.




$v_1 \times v_2 < 0$ $\qquad$ $v_1 \times v_2 > 0$

Notice that if a point $q$ is on the right, then the cross product $v_1 \times v_2$ will be negative, or if $q$ is on the left, it will be positive. If $q$ happens to be on the ray, then the cross product is zero.

***Algorithm: Line-side test***

```
let sign(x) = if x<0 then −1 else if x>0 then 1 else 0

let line-side-test(p1, p2, q)=
   let v1 = p2 − p1
   let v2 = q − p1
   return sign(v1 × v2)
```

# 3 The Chop Problem

One of the challenges of this problem is to characterize the set of moves for Chopper. Such a move partitions the points of the input set into two non-empty parts, separated by a straight line. There is an elegant bijection between unordered pairs of distinct points with such a partition. This means that there are exactly $\binom{n}{2}$ moves for Chopper. This is something you'll have to figure out.

By the way, $\binom{n}{2}$ is also the number of different moves that Linker can make.