CONVEX HULLS

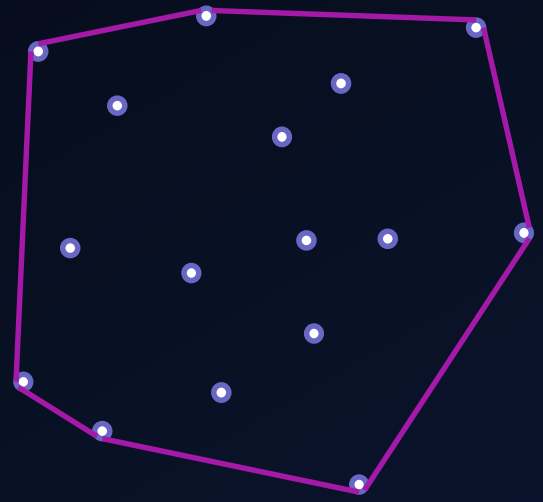
- Convex hull determination
- Graham scan algorithm
- Jarvis march (Gift wrapping)

Author: prof. Yevhenii Borodavka

Convex hull determination

A convex hull of a given set of points is the smallest convex polygon containing the points. Intuitively, the convex hull is what you get by driving a nail into the plane at each point and then wrapping a piece of string around the nails. In the figure below, figure (a) shows a set of points, and figure (b) shows the corresponding convex hull.

(a)



(b)

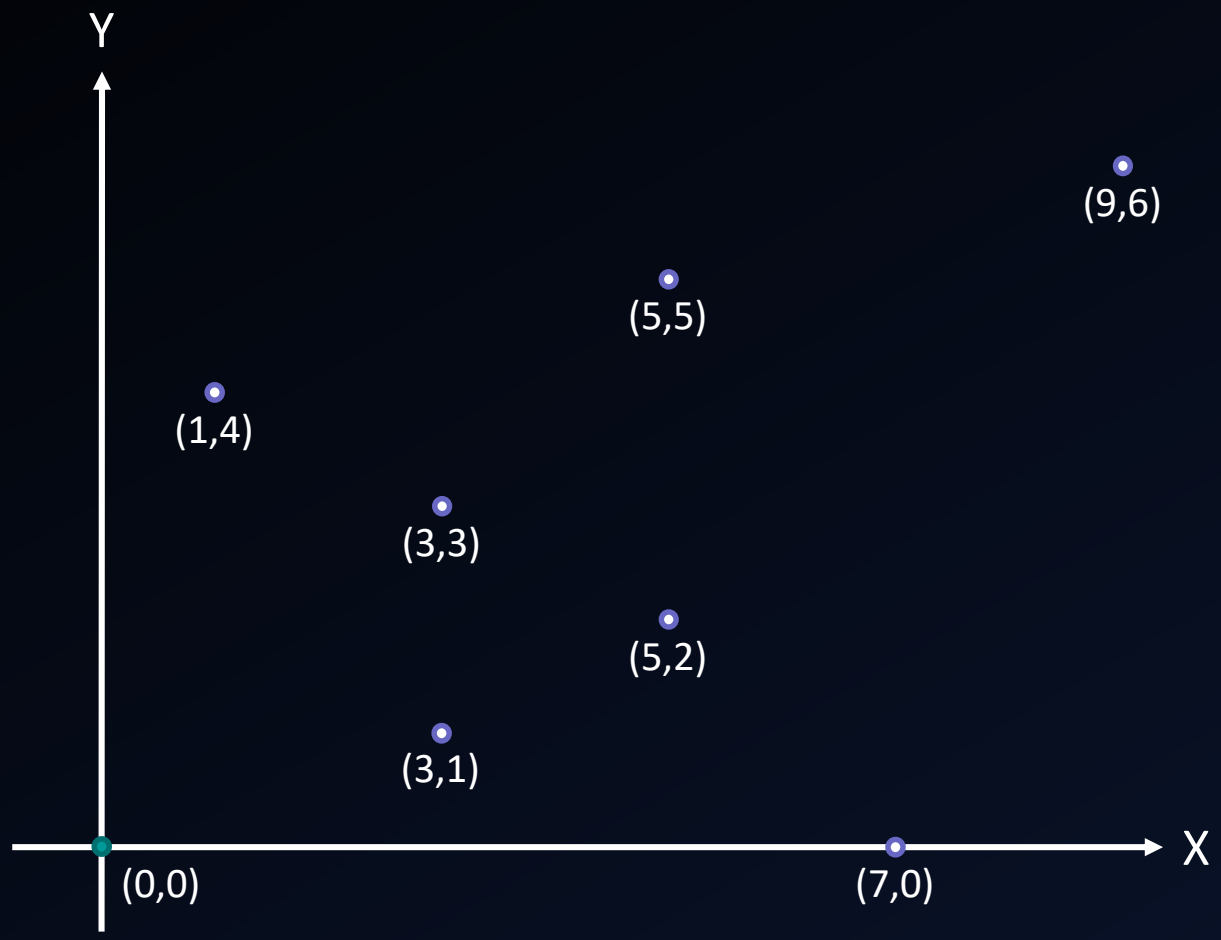
Graham scan algorithm

Graham scan is an algorithm to compute a convex hull of a given set of points in O(n*log(n)) time. This algorithm first sorts the set of points according to their polar angle and scans the points to find the convex hull vertices. The algorithm steps are below.

1. Find the point (P_0) with the smallest y-coordinate. In case of a tie, choose the point with the smallest x-coordinate. This step takes O(n) time.
2. Sort the points based on the polar angle i.e. the angle made by the line with the X-axis. While implementing, we don't calculate the angle, instead, we calculate the relative orientation of two points to find out which point makes the larger angle. We can use any sorting algorithm that has complexity O(n*log(n)).
3. After sorting, we check for the collinear points. If we find any collinear points, we keep the furthest point from P_0 and remove all other points. This step takes O(n) time.
4. The first two points in the sorted list are always in the convex hull. We maintain a stack data structure to keep track of the convex hull vertices. We push these two points and the next point in the list to the stack.
5. Now we check if the next point in the list turns left or right from the two points on the top of the stack. If it turns left, we push this item on the stack. If it turns right, we remove the item on the top of the stack and repeat this process for the remaining items. This step takes O(n) time.

Graham scan algorithm

An example. STEP 1. The green color means the point is in the convex hull and the red color means the point can not be in the convex hull.



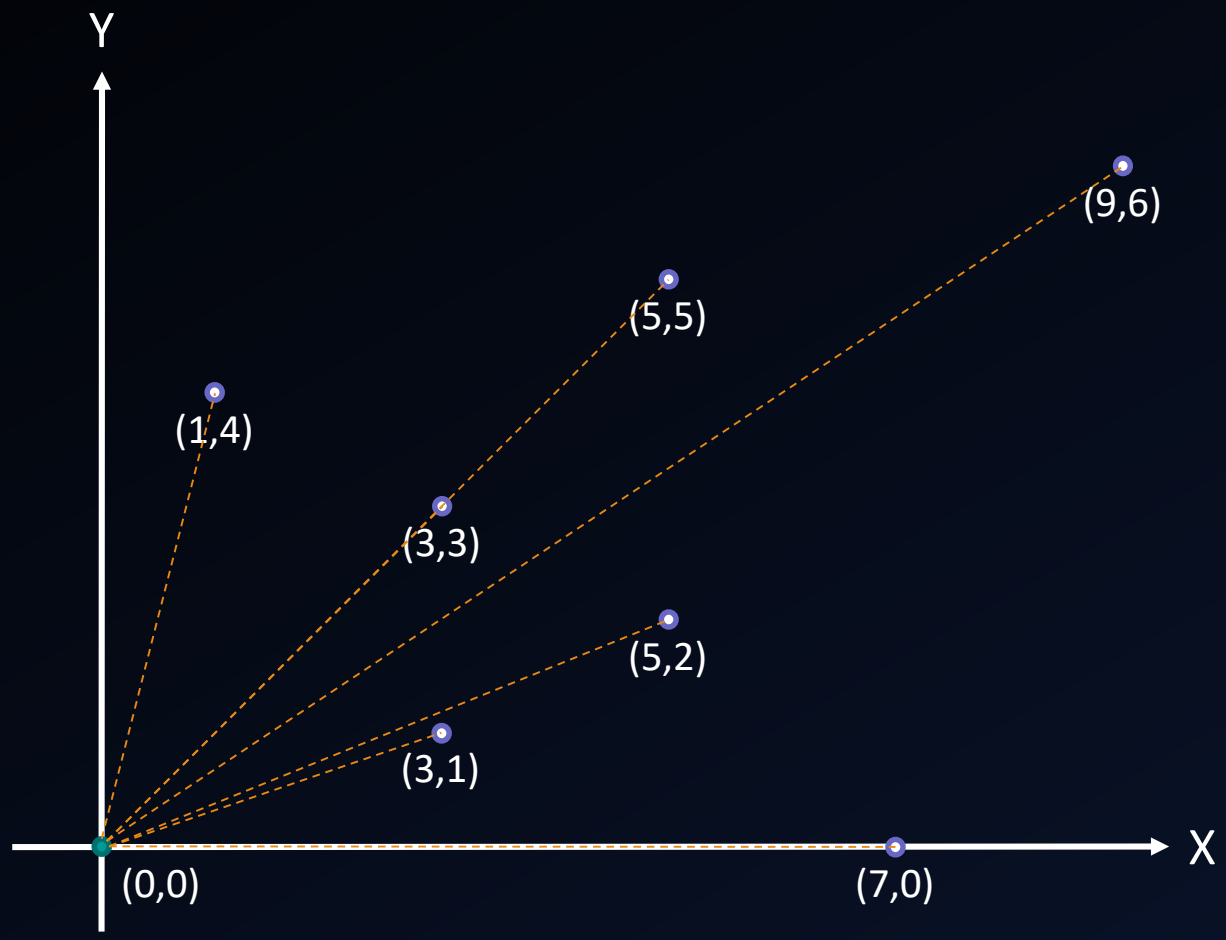
The program first finds the point with the smallest y-coordinate. There are two candidate points for this (0,0) and (0,7). Since this is a tie, the program chooses the one with a smaller x-coordinate which is (0,0). This point is guaranteed to be in the convex hull.

Cross-product of two vectors with the three points (p_1,p_2,p_3) is

$(x_2-x_1)*(y_3-y_1)-(x_3-x_1)*(y_2-y_1)$

GRAHAM SCAN ALGORITHM

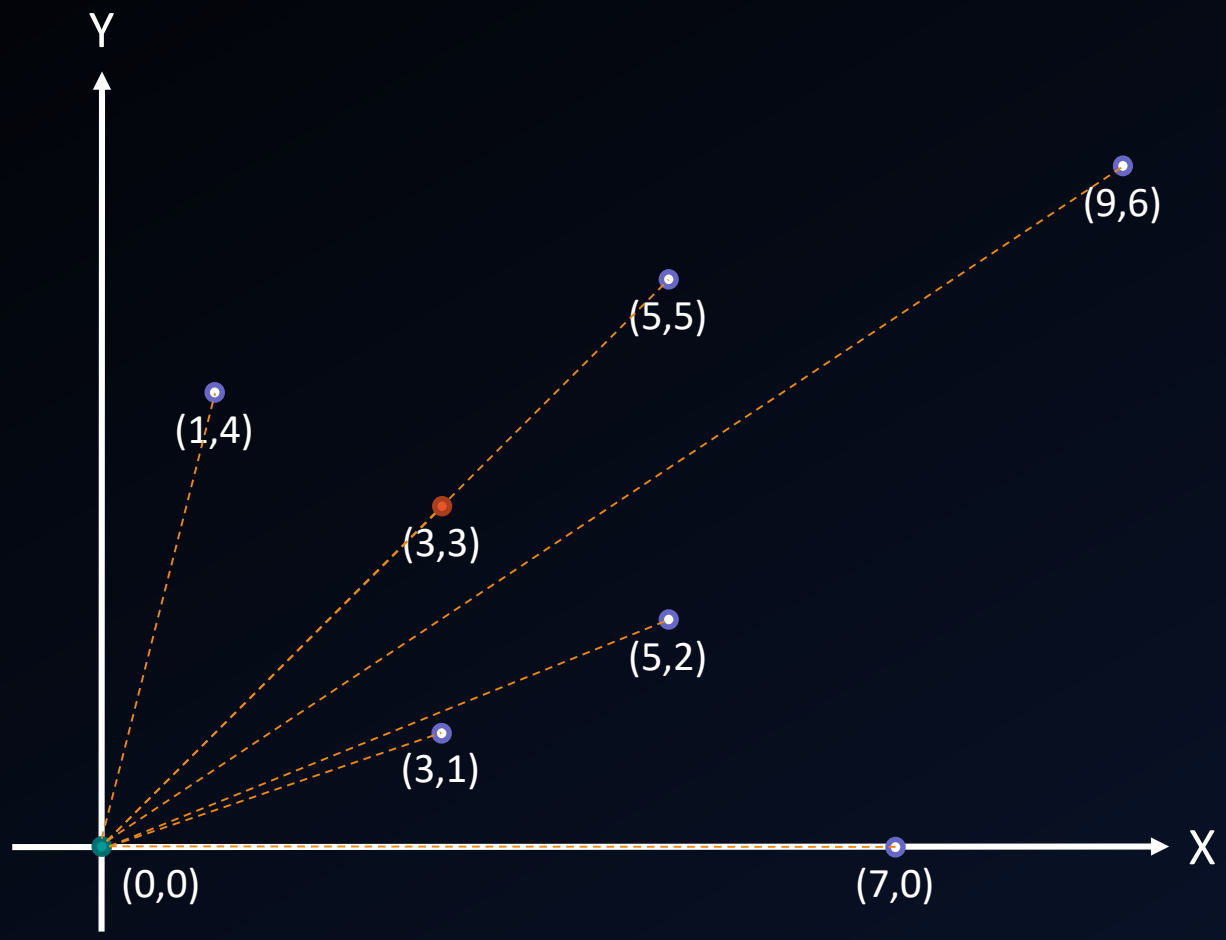
An example. STEP 2. The green color means the point is in the convex hull and the red color means the point can not be in the convex hull.



The program sorts the points based on the polar angle. To find out whether the line (0,0)-(5,2) or the line (0,0)-(3,1) makes the larger angle with the X-axis, we calculate the cross-product of vector (0,0)-(3,1) and vector (0,0)-(5,2). If the cross-product is positive, that means the vector (0,0)-(3,1) clockwise from the vector (3,1)-(5,2) with respect to the X-axis. This indicates that the angle made by the vector (0,0)-(5,2) is larger than the angle made by the vector (0,0)-(3,1).
The sorted points are [(0,0), (7,0), (3,1), (5,2), (9,6), (3,3), (5,5), (1,4)].

GRAHAM SCAN ALGORITHM

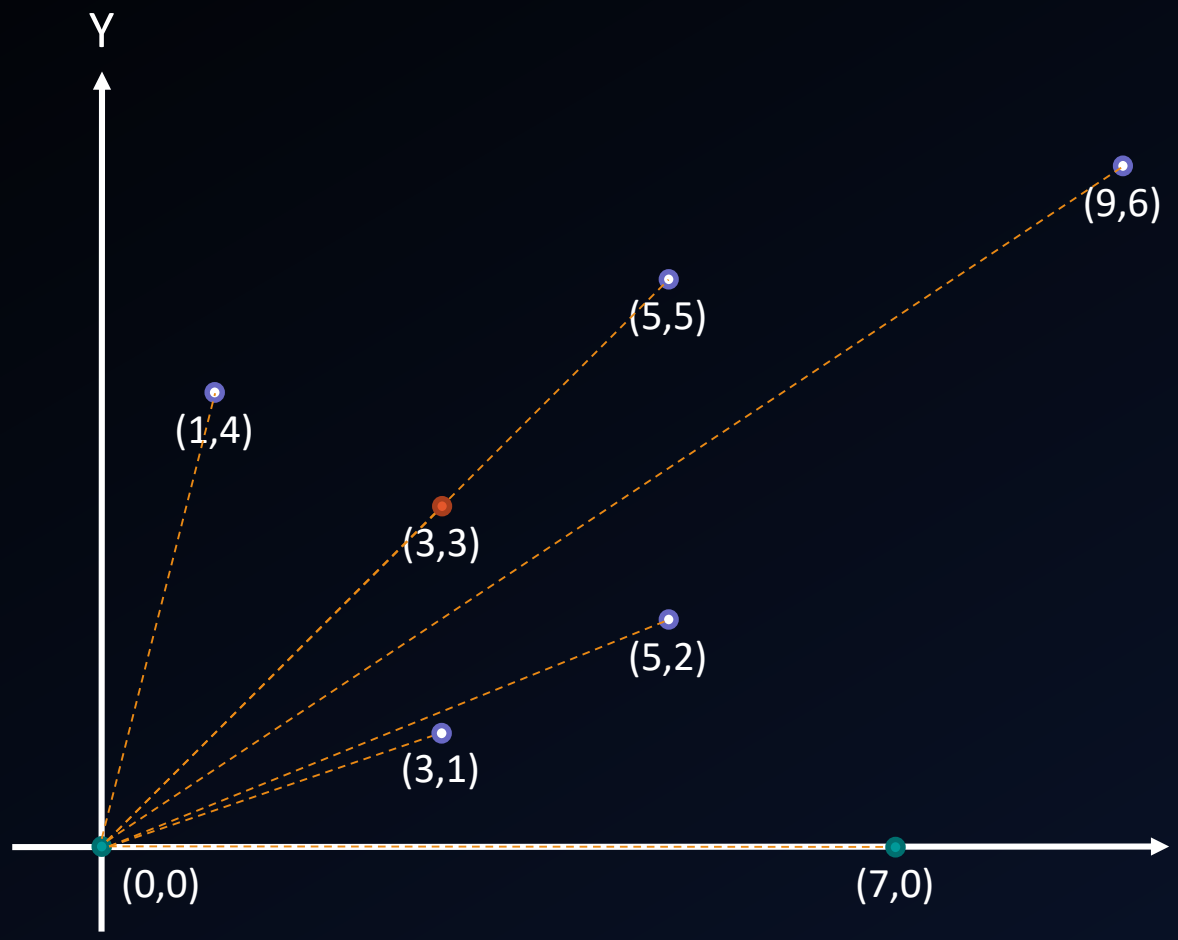
An example. STEP 3. The green color means the point is in the convex hull and the red color means the point can not be in the convex hull.



Next, it searches for the collinear points and keeps the farthest point. If the cross-product of two vectors is zero then we have collinear points. Here points (3,3) and (5,5) are collinear with (0,0). Point (5,5) is kept and (3,3) is discarded as (5,5) is far from (0,0). For distance comparison, we use the square distance between two points: $(x_1 - x_2)^2 + (y_1 - y_2)^2$.

GRAHAM SCAN ALGORITHM

An example. STEP 4. The green color means the point is in the convex hull and the red color means the point can not be in the convex hull.

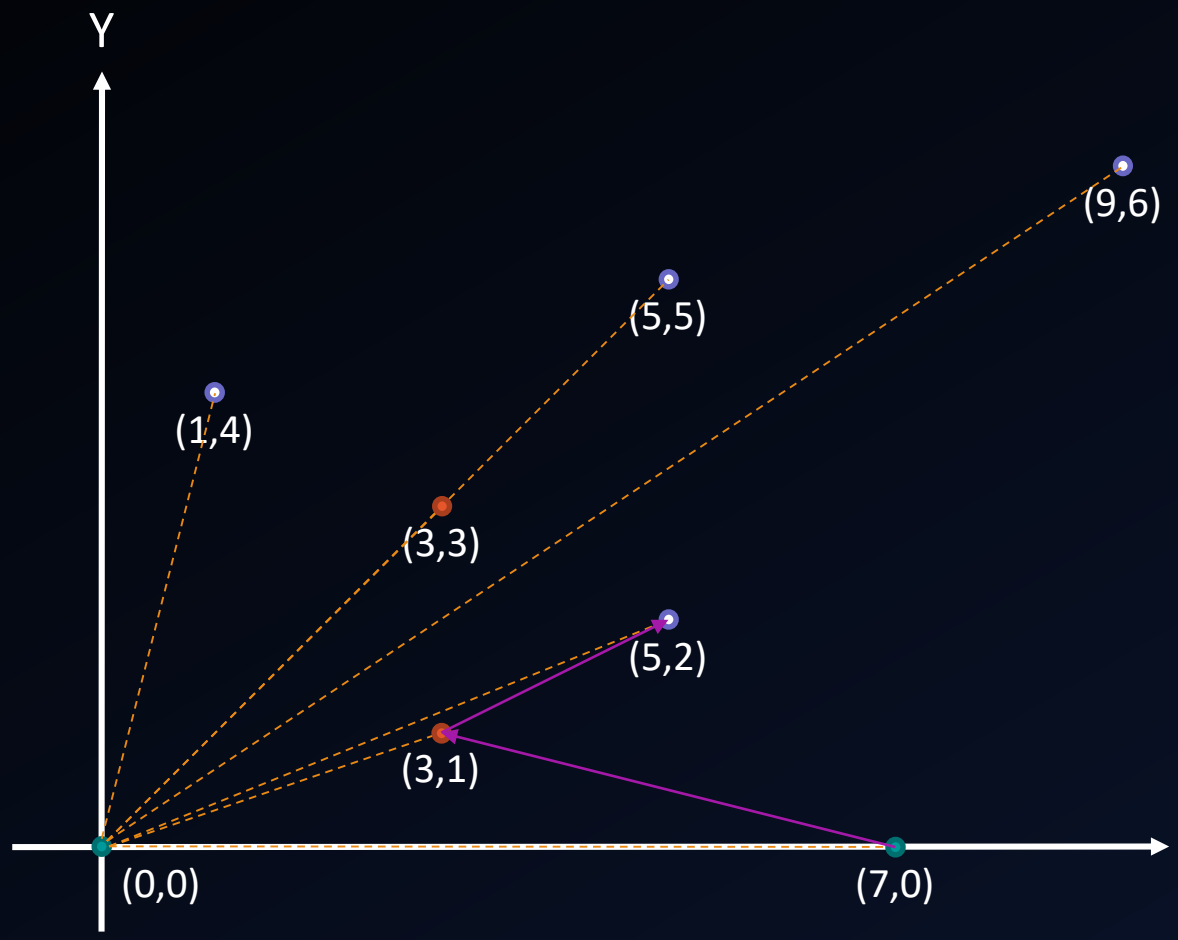


Next, the program pushes the first three points from the sorted list to the stack. The first two points are always in the convex hull.

(3,1)	(7,0)	(0,0)	

GRAHAM SCAN ALGORITHM

An example. STEP 5. The green color means the point is in the convex hull and the red color means the point can not be in the convex hull.

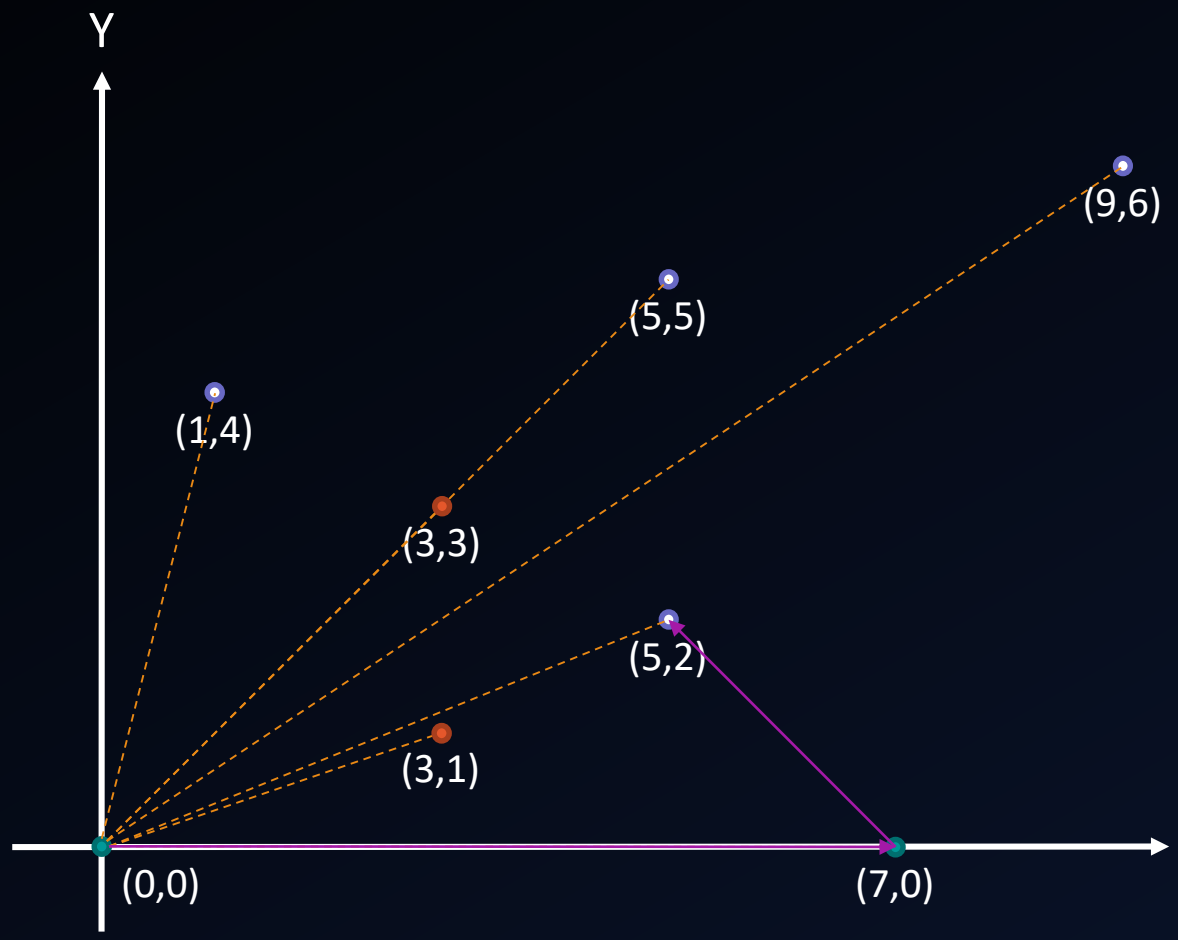


Next, it checks if the next point in the list turns right or left from the two top points in the stack. In this case, it checks if point (5,2) turns left or right from point (7,0) and (3,1). Since this is a right turn, the point (3,1) is popped from the stack as it can not be in the convex hull.

~~(3,1)~~	(7,0)	(0,0)	

GRAHAM SCAN ALGORITHM

An example. STEP 6. The green color means the point is in the convex hull and the red color means the point can not be in the convex hull.

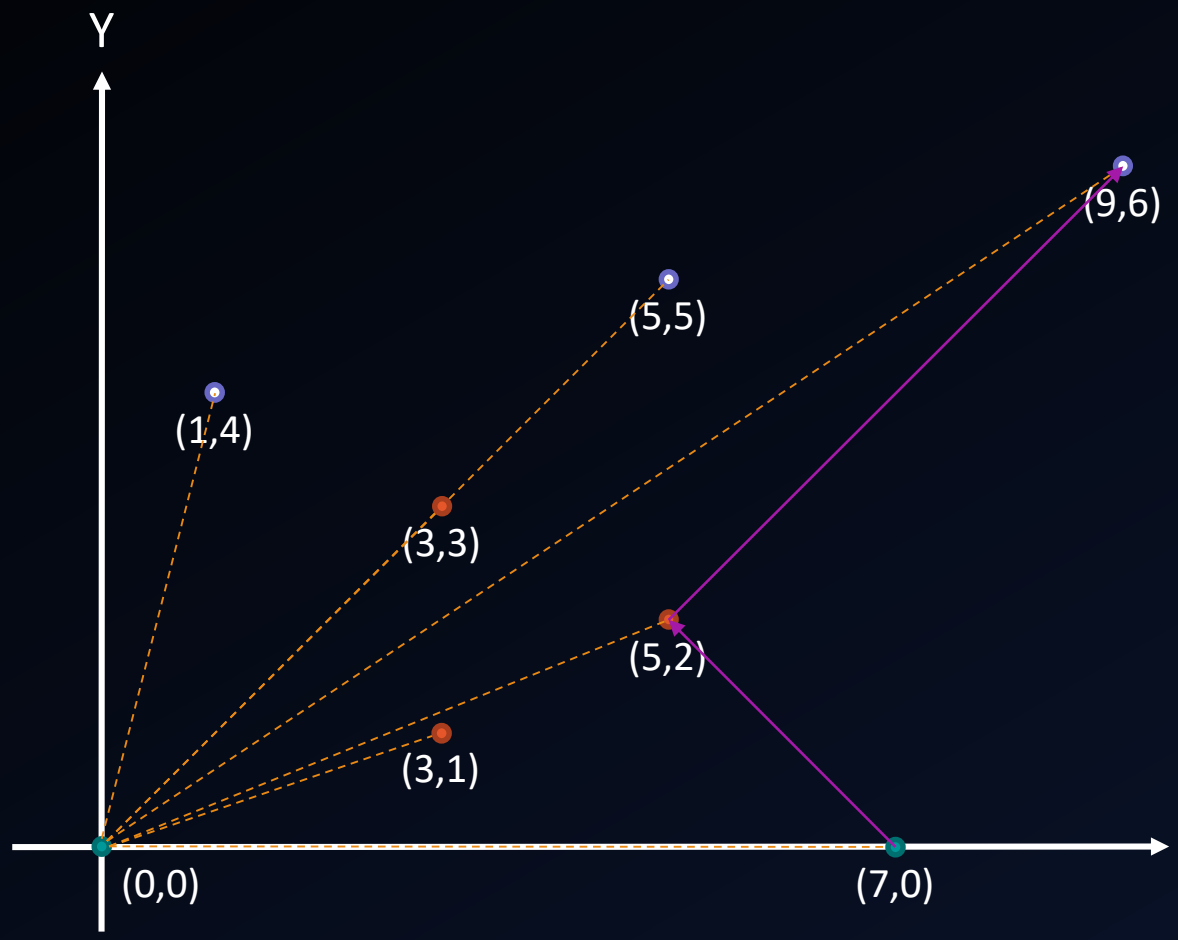


Similarly, it checks if the new point in the list (5,2) turns left or right from points (0,0) and (7,0). It turns left, so the point is pushed to the stack.

(5,2)	(7,0)	(0,0)	

GRAHAM SCAN ALGORITHM

An example. STEP 7. The green color means the point is in the convex hull and the red color means the point can not be in the convex hull.

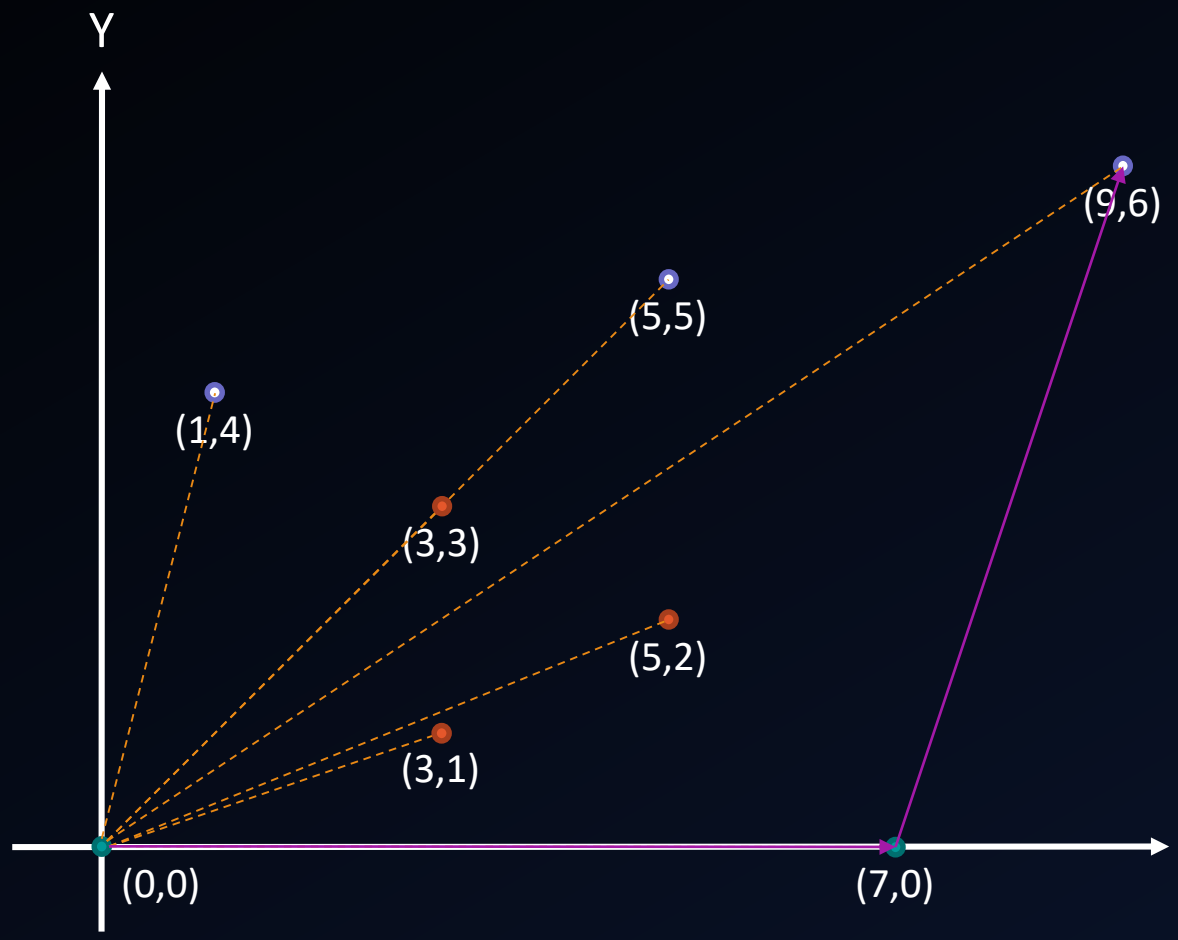


The same process goes on. The next point is (9,6). It makes a left turn, so we discard point (5,2).

~~(5,2)~~	(7,0)	(0,0)	

GRAHAM SCAN ALGORITHM

An example. STEP 8. The green color means the point is in the convex hull and the red color means the point can not be in the convex hull.

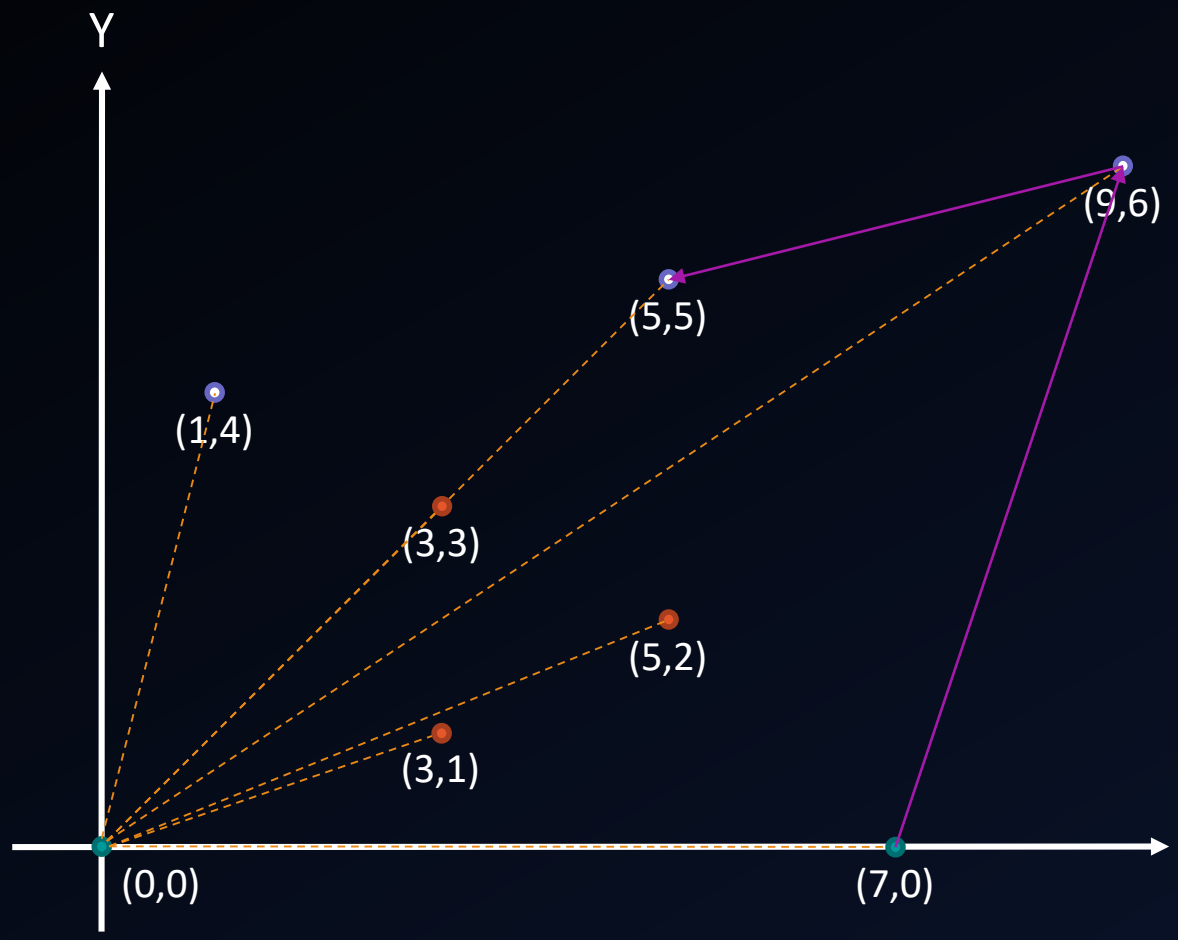


The next point (9,6) is pushed into the stack.

(9,6)	(7,0)	(0,0)	

GRAHAM SCAN ALGORITHM

An example. STEP 9. The green color means the point is in the convex hull and the red color means the point can not be in the convex hull.

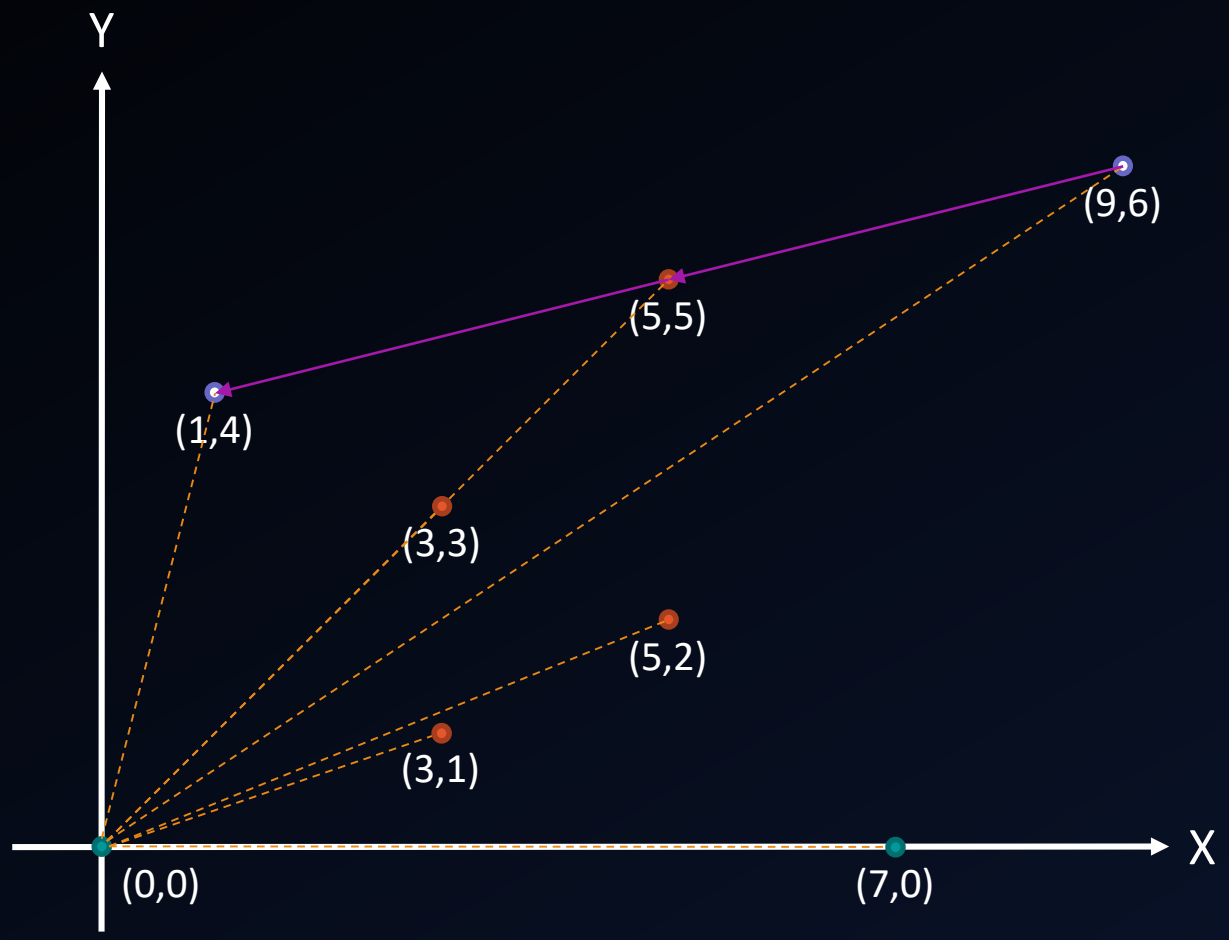


In the same way (5,5) is pushed into the stack.

(5,5)	(9,6)	(7,0)	(0,0)

GRAHAM SCAN ALGORITHM

An example. STEP 10. The green color means the point is in the convex hull and the red color means the point can not be in the convex hull.

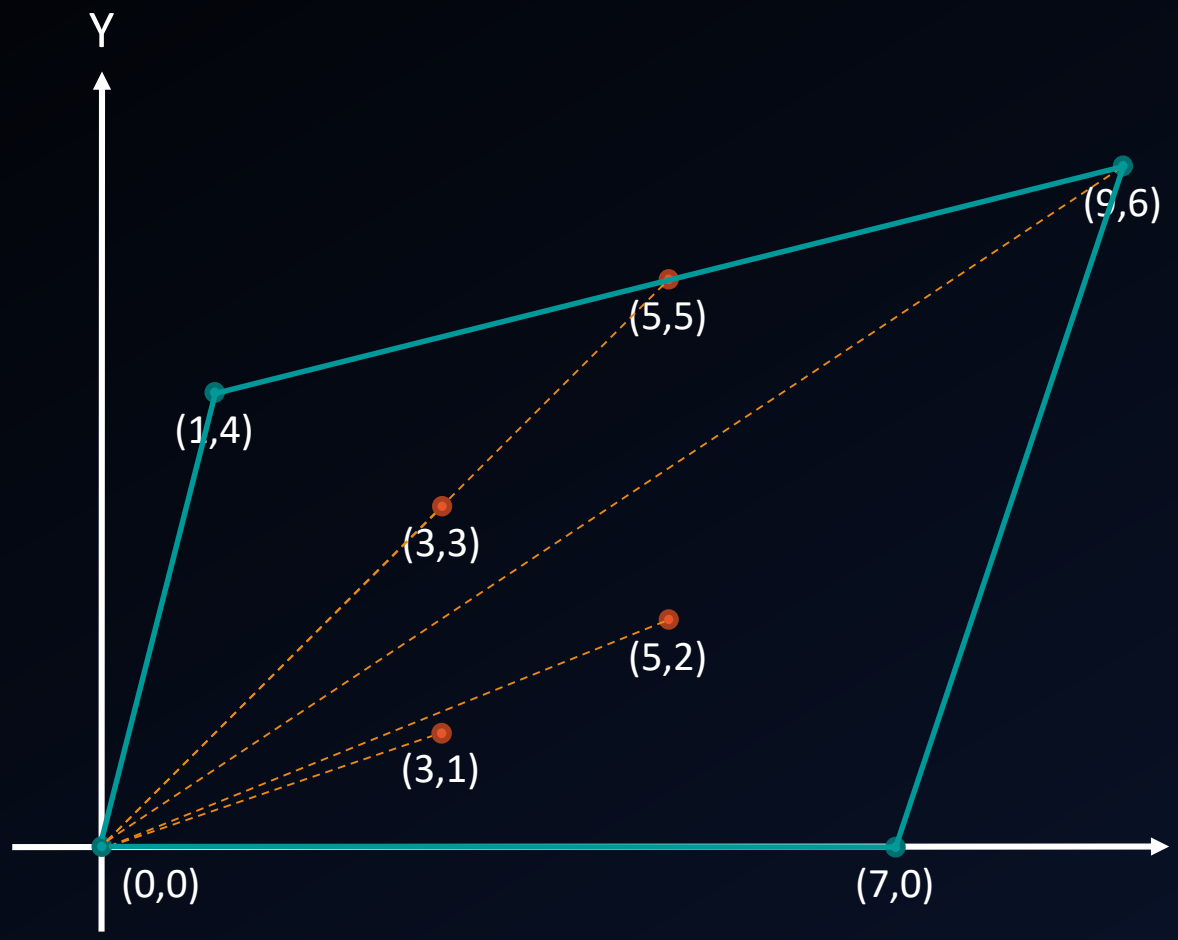


Next, point (1,4) is collinear with points (5,5) and (9,6). In the case of collinearity, we discard the top of the stack. Point (5,5) is popped from the stack.

~~(5,5)~~	(9,6)	(7,0)	(0,0)

GRAHAM SCAN ALGORITHM

An example. STEP 10. The green color means the point is in the convex hull and the red color means the point can not be in the convex hull.



Next, point (1,4) is pushed into the stack. Since point (1,4) is the last point in the list, the algorithm terminates here. The points in the stack are the convex hull.

(1,4)	(9,6)	(7,0)	(0,0)

Jarvis march (Gift wrapping)

Jarvis's march algorithm uses a process called *gift wrapping* to find the convex hull. It is one of the simplest algorithms for computing the convex hull. The working of Jarvis's march resembles the working of selection sort. In selection sort, in each pass, we find the smallest number and add it to the sorted list. Similarly, in Jarvis's march, we find the *leftmost point* and add it to the convex hull vertices in each pass.

The algorithm spends **O(n)** time on each convex hull vertex. If there are **h** convex hull vertices, the total time complexity of the algorithm would be **O(nh)**. Since **h** is the number of outputs of the algorithm, this algorithm is also called an *output-sensitive* algorithm since the complexity also depends on the number of outputs.

Jarvis march (Gift wrapping)

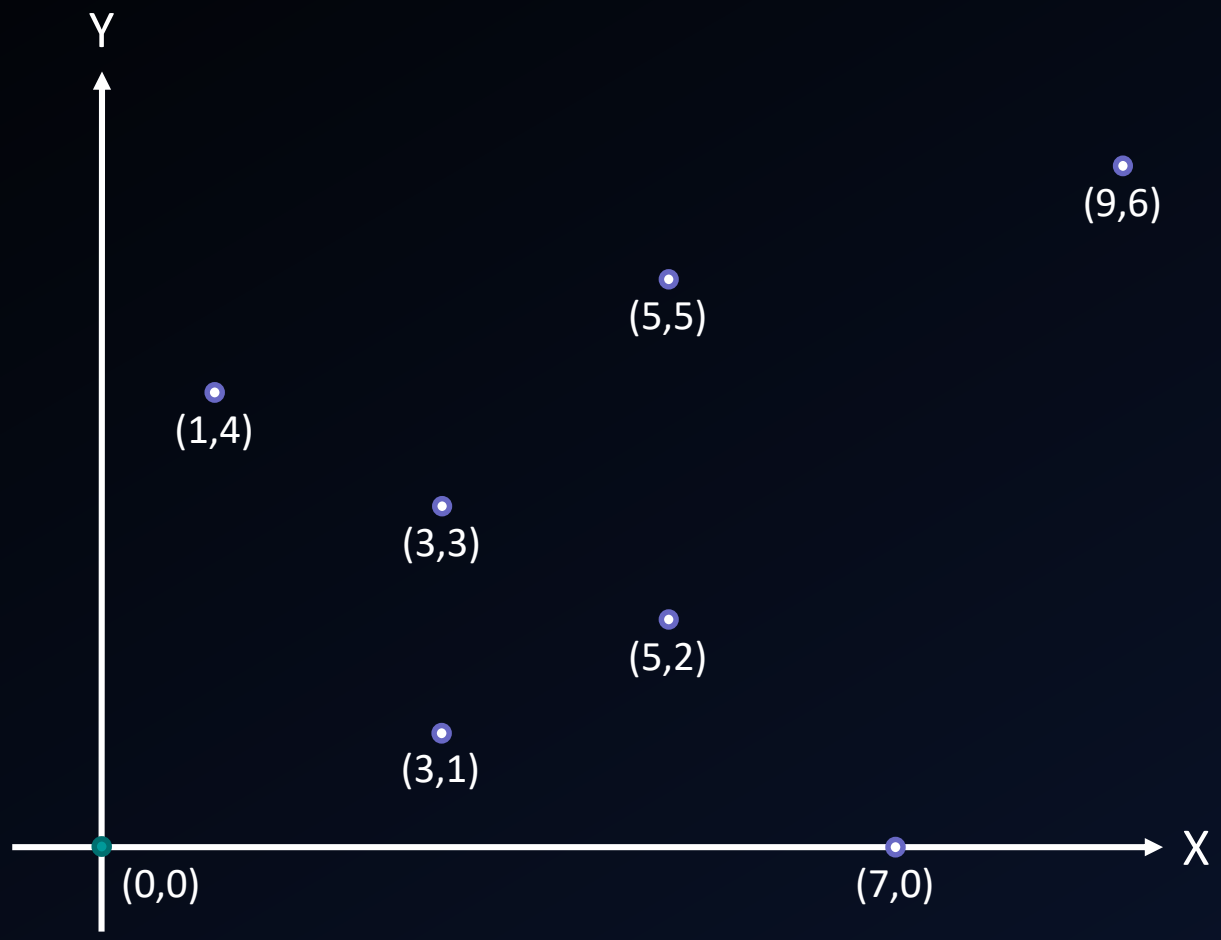
The step-by-step working of Jarvis's march algorithm is given below.

1. From the given set of points **P**, we find a point with minimum x-coordinates (or leftmost point with reference to the x-axis). Let's call this point **p**. Since this point is guaranteed to be in the convex hull, we add this point to the list of convex hull vertices.
2. From **p**, find the leftmost point. For this, we do the following. We select the vertex following **p** and call it **q**. We check if **q** is turning right from the line joining **p** and every other point one at a time. If **q** is turning right, we move **q** to the point from where it was turning right. This way we move **q** towards the left in each iteration and finally stop when **q** is in the leftmost position from **p**. We add **q** to the list of convex hull vertices.
3. Now **q** becomes **p** and we repeat the step (2).
4. Repeat steps (2) and (3) until we reach the point where we started.

After the execution of this algorithm, we should get the correct convex hull.

JARVIS MARCH (GIFT WRAPPING)

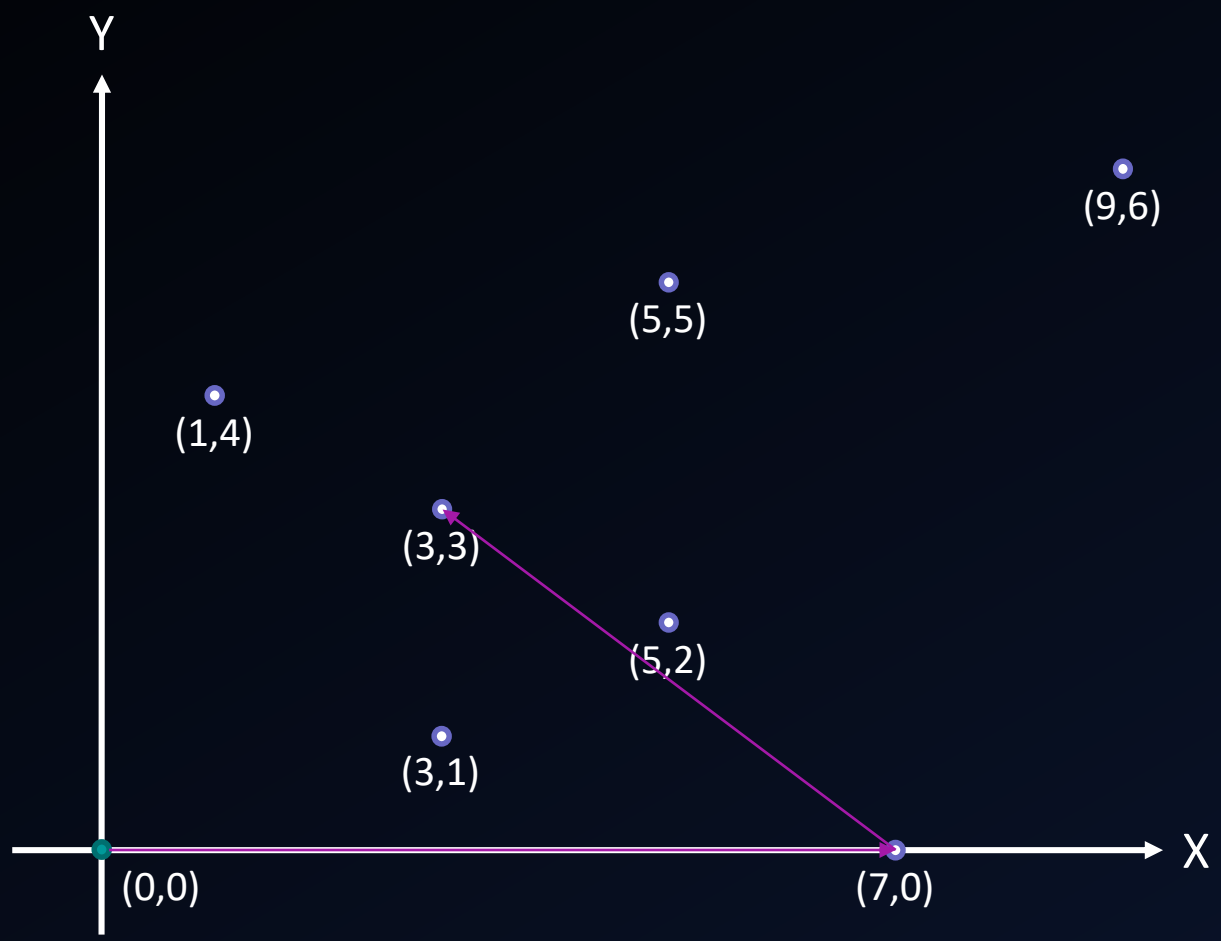
An example. STEP 1. The green color means the point is in the convex hull.



The program first finds the leftmost point by sorting the points on x-coordinates. The leftmost point for the above set of points is **p**=(0,0). We insert the point (0,0) into the convex hull vertices as shown by the green circle in the figure left. Next, we find the leftmost point from point **p**=(0,0).

JARVIS MARCH (GIFT WRAPPING)

An example. STEP 2.1. The green color means the point is in the convex hull.



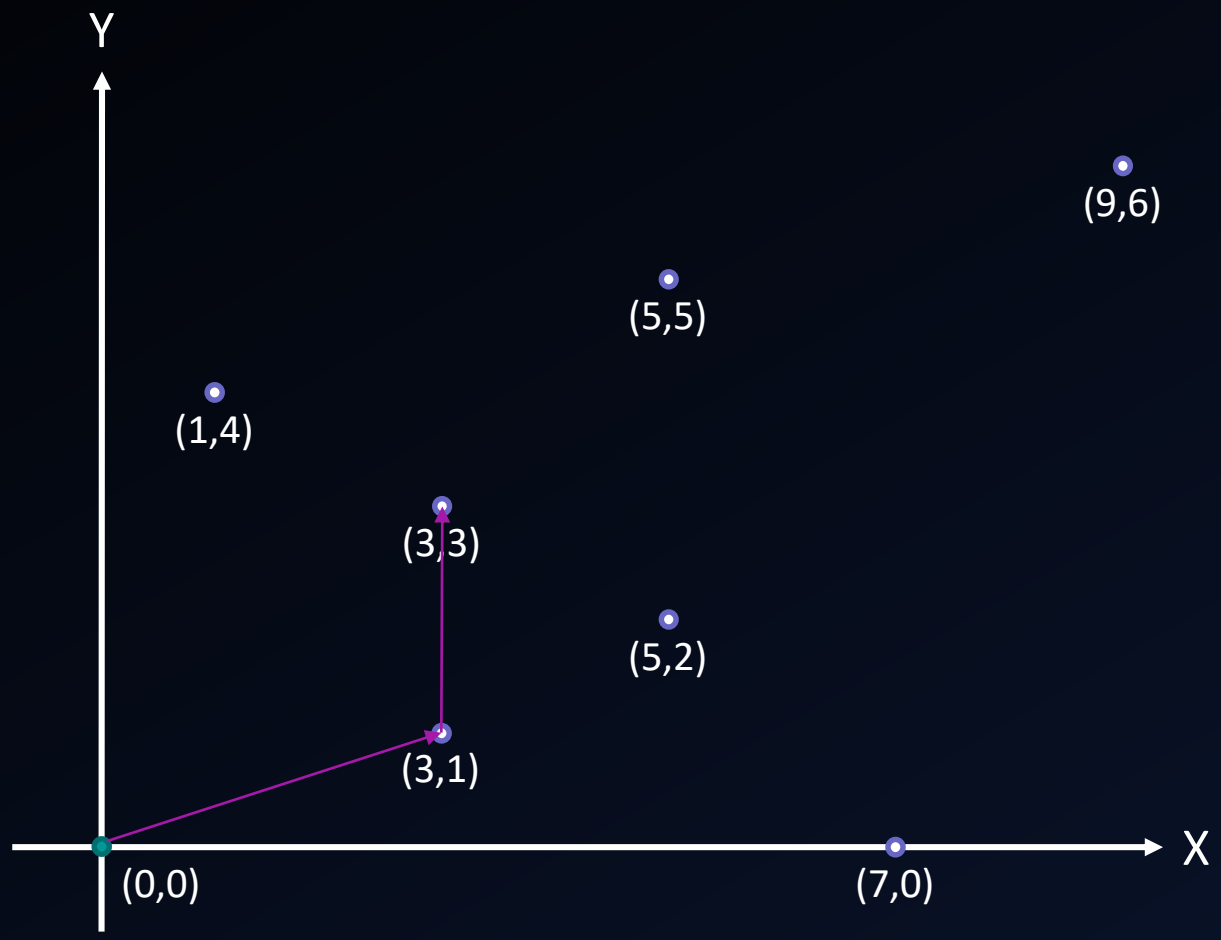
We pick a point following **p** and call it **q**. Let q be the point (3,3).

Let all other points except **p** and **q** be **j**. Now we check whether the sequence of points (p,j,q) turns right. If it turns right, we replace **q** with **j** and repeat the same process for the remaining points.

Let **j**=(7,0). The sequence [(0,0),(7,0),(3,3)] turns left. Since we only care about the right turn, we don't do anything in this case and simply move on.

JARVIS MARCH (GIFT WRAPPING)

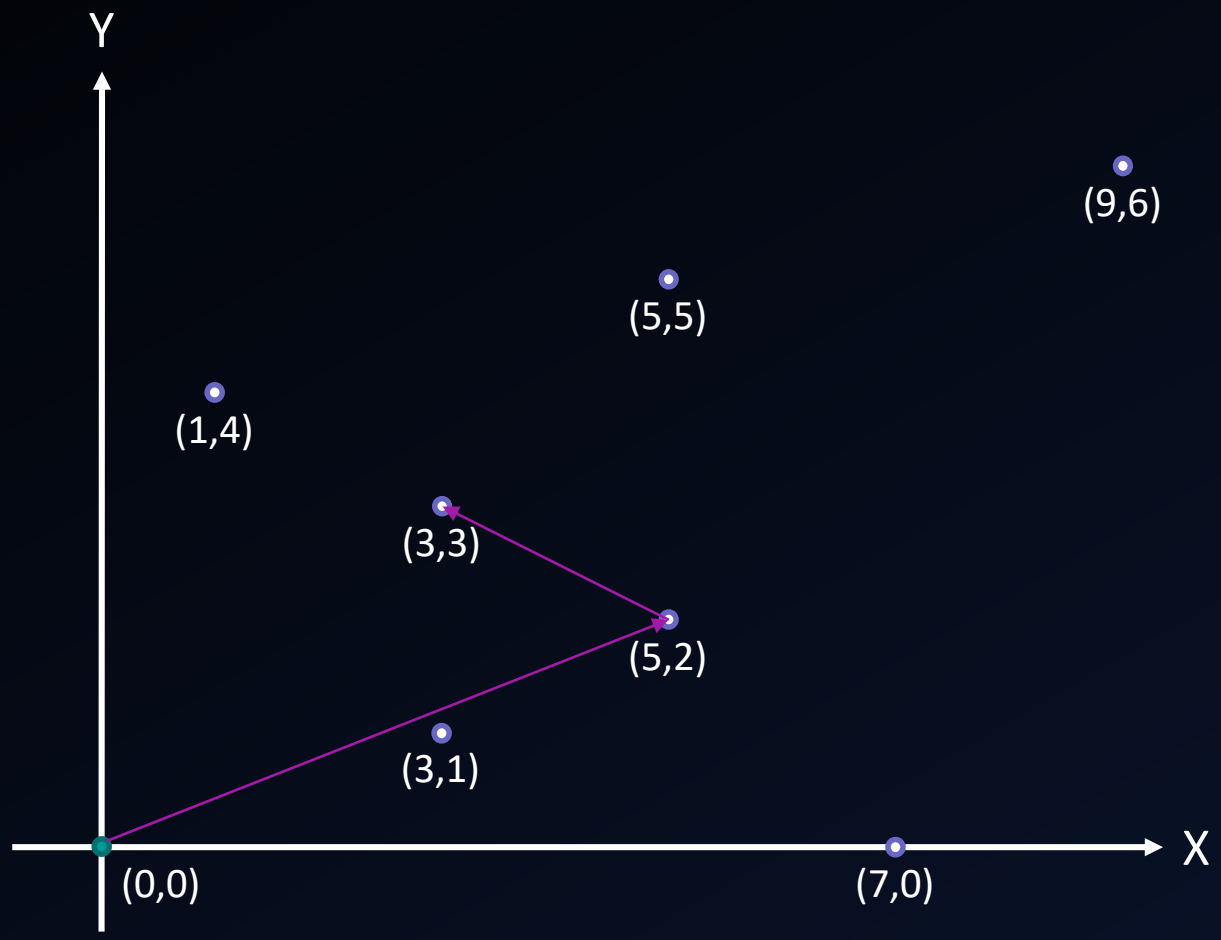
An example. STEP 2.2. The green color means the point is in the convex hull.



Let **j**=(3,1). The sequence [(0,0),(3,1),(3,3)] turns left and we move on without doing anything.

JARVIS MARCH (GIFT WRAPPING)

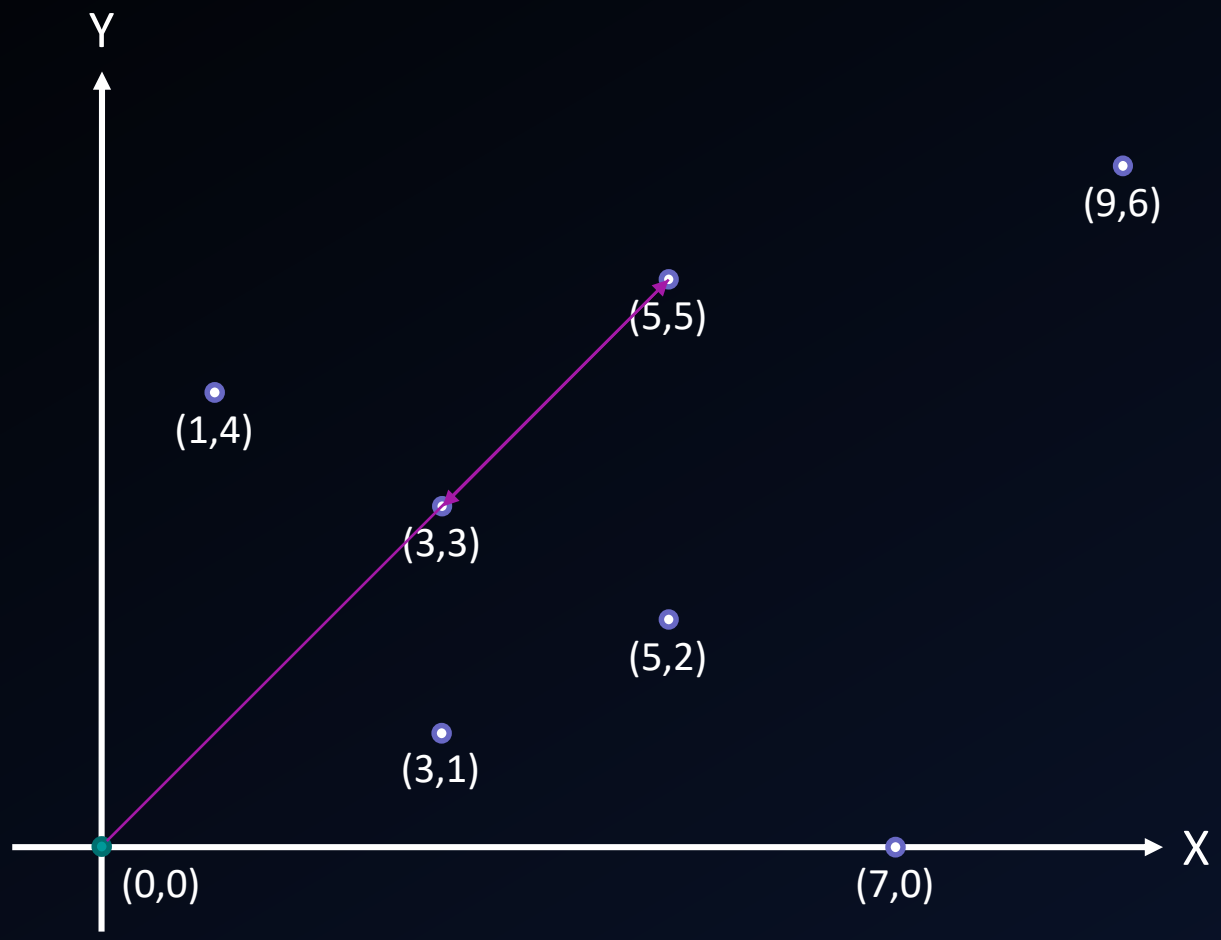
An example. STEP 2.3. The green color means the point is in the convex hull.



Let **j**=(5,2). The sequence [(0,0),(5,2),(3,3)] turns left and we move on without doing anything.

JARVIS MARCH (GIFT WRAPPING)

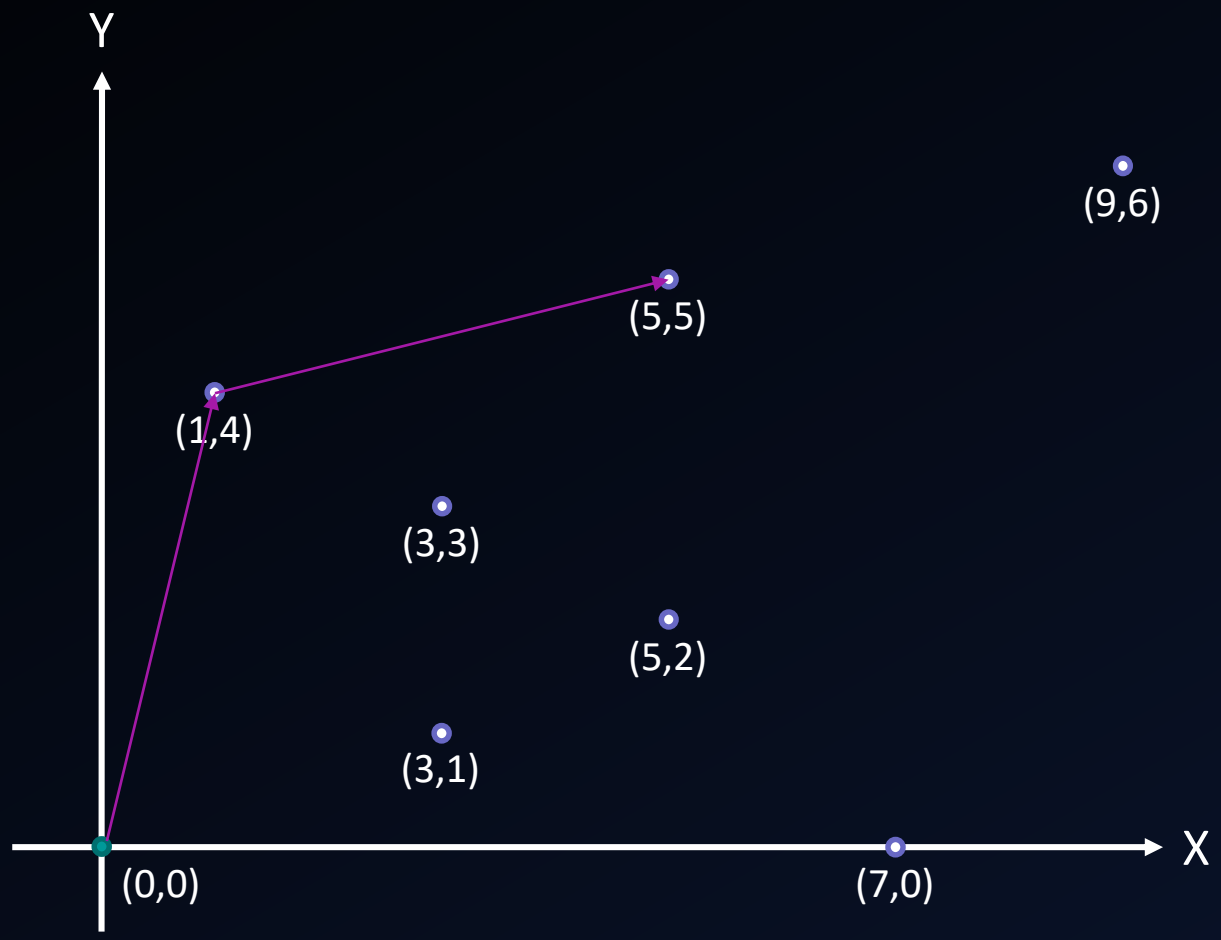
An example. STEP 2.4. The green color means the point is in the convex hull.



Let next **j**=(5,5). The sequence [(0,0),(5,5),(3,3)] is collinear. In the case of collinear, we replace **q** with **j** only if the distance between **p** and **j** is greater than the distance between **q** and **p**. In this case, the distance between (0,0) and (5,5) is greater than the distance between (0,0) and (3,3) we replace **q** with point (5,5).

Jarvis march (Gift wrapping)

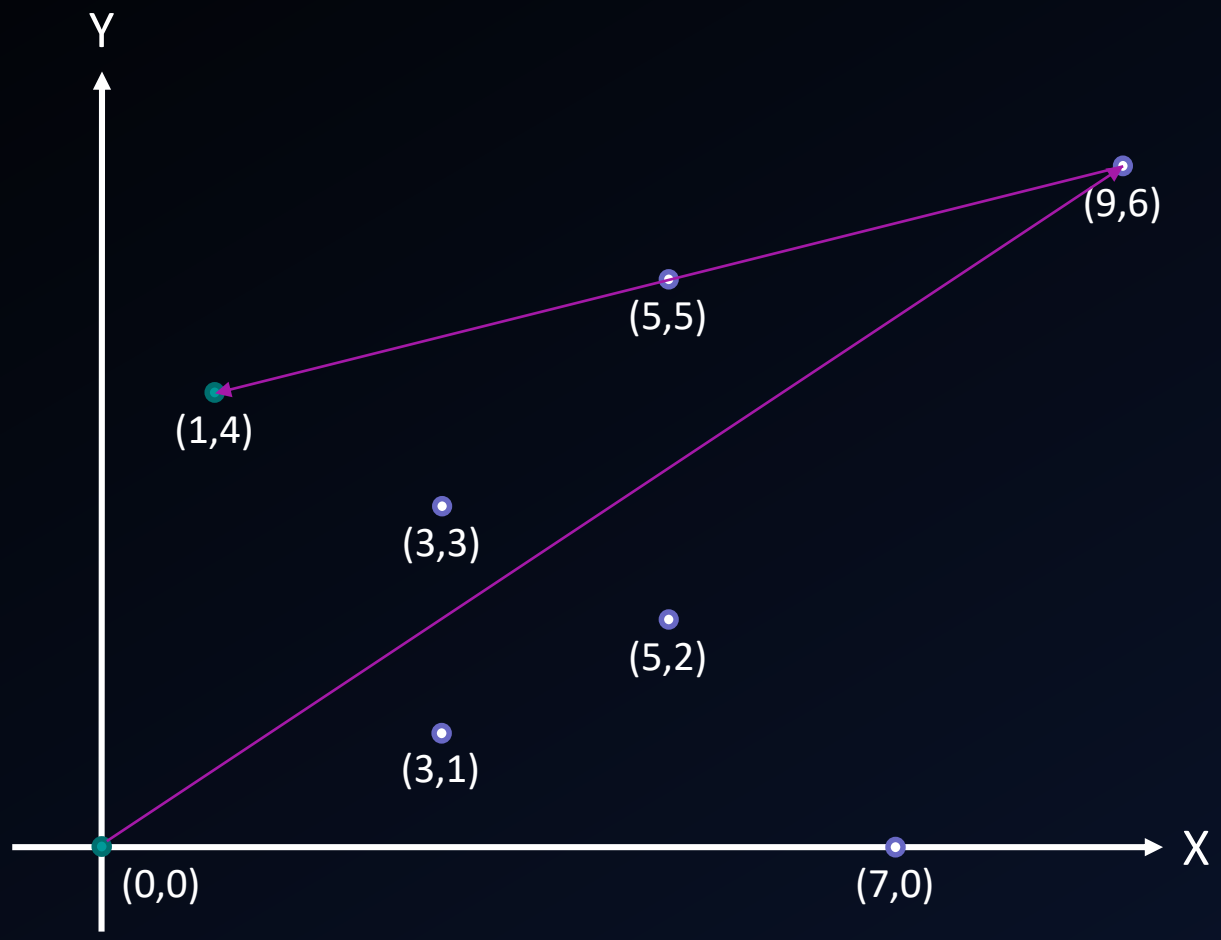
An example. STEP 2.5. The green color means the point is in the convex hull.



Let next **j**=(1,4). The sequence [(0,0),(1,4),(5,5)] turns right. We replace **q** with point (1,4).

Jarvis march (Gift wrapping)

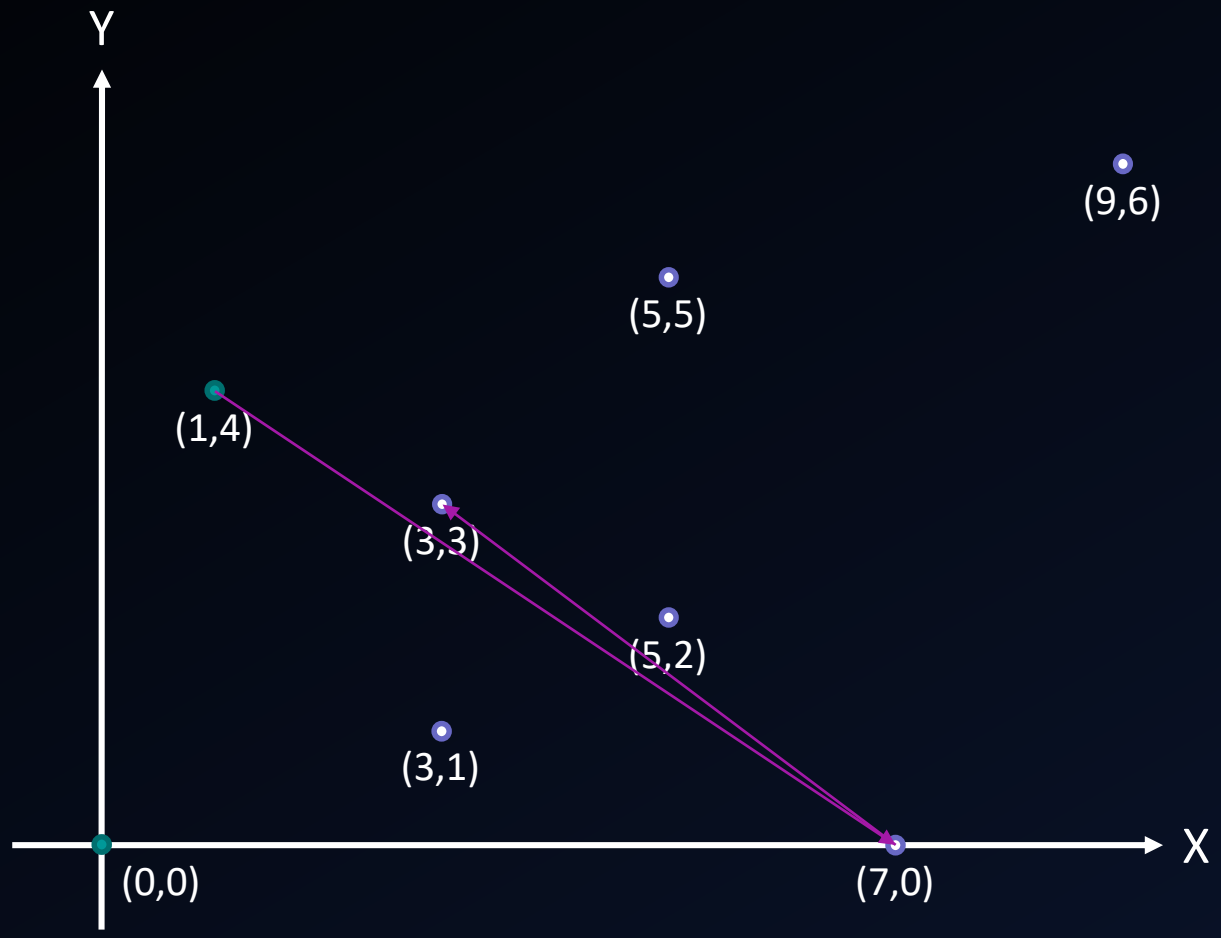
An example. STEP 2.6. The green color means the point is in the convex hull.



Finally, the last choice for **j** is (9,6). The sequence [(0,0),(9,6),(1,4)] turns left. So we do nothing. We went through all the points and now **q**=(1,4) is the leftmost point.
We add point (1,4) to the convex hull.

Jarvis march (Gift wrapping)

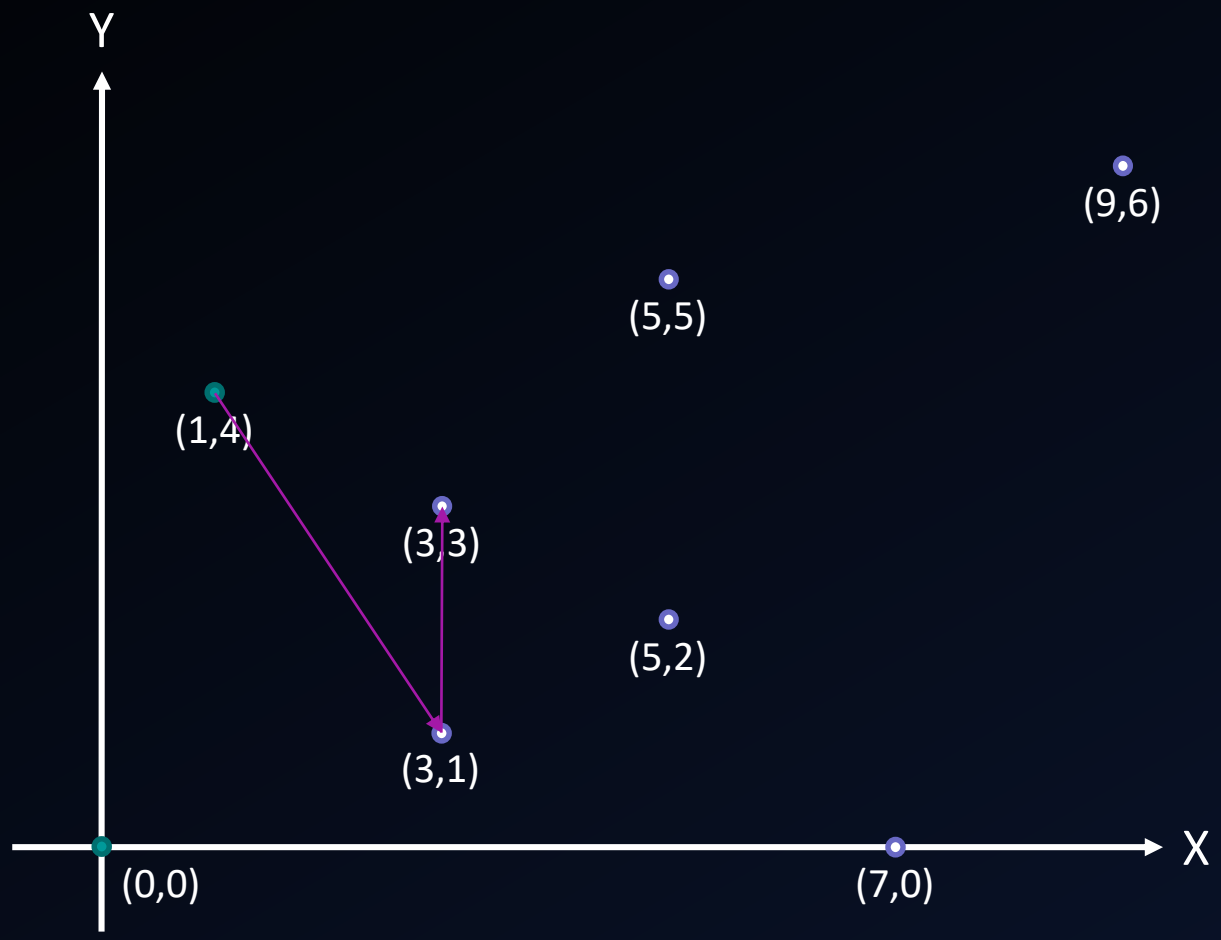
An example. STEP 3.1. The green color means the point is in the convex hull. The previously added points checking are not shown during this step.



Next, we find the leftmost point from the point (1,4) to the convex hull.
Let **q** be the point (3,3).
Let **j**=(7,0). The sequence [(1,4),(7,0),(3,3)] turns left and we move on without doing anything.

Jarvis march (Gift wrapping)

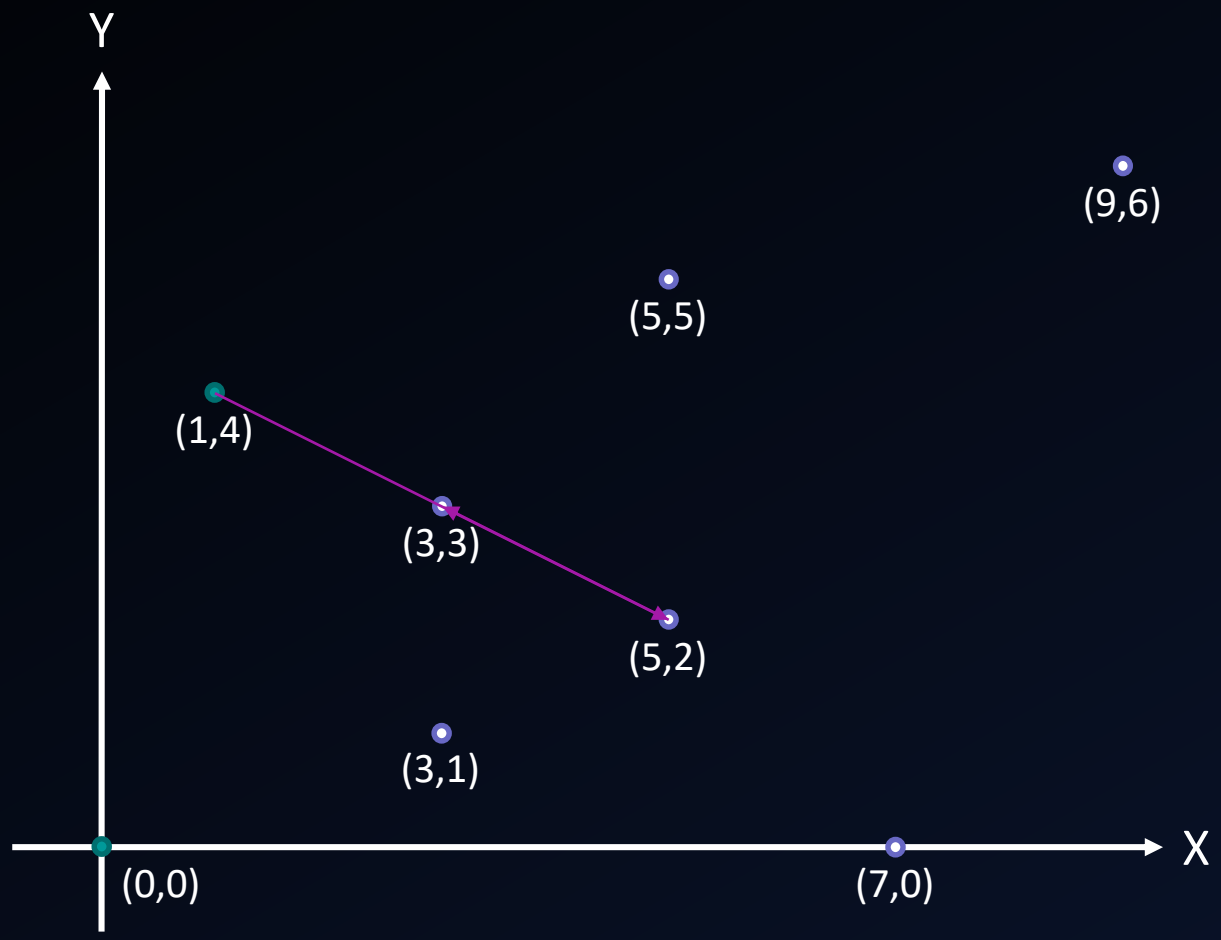
An example. STEP 3.2. The green color means the point is in the convex hull.



Let **j**=(3,1). The sequence [(1,4),(3,1),(3,3)] turns left and we move on without doing anything.

JARVIS MARCH (GIFT WRAPPING)

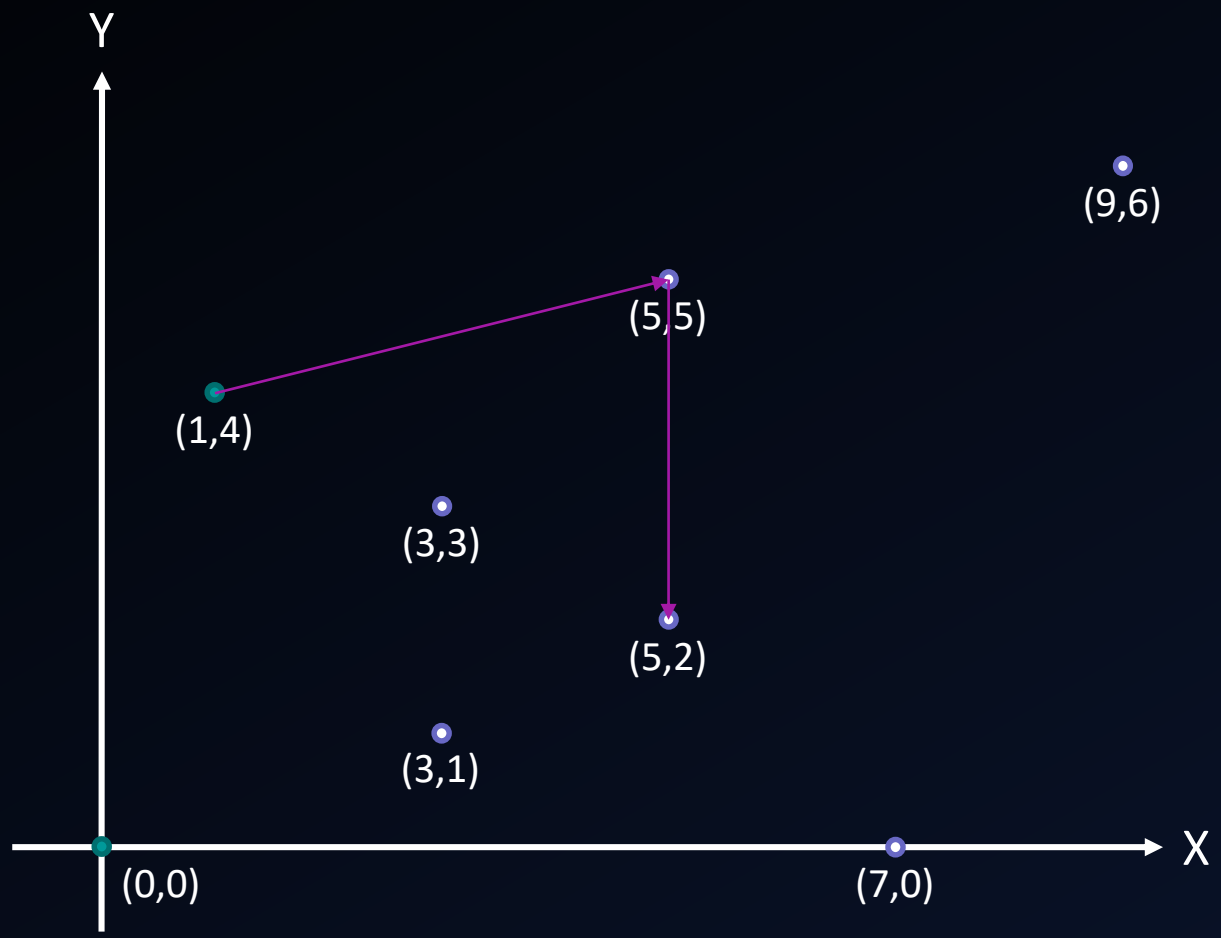
An example. STEP 3.3. The green color means the point is in the convex hull.



Let **j**=(5,2). The sequence [(1,4),(5,2),(3,3)] is collinear. In the case of collinear, we replace **q** with **j** only if the distance between **p** and **j** is greater than the distance between q and **p**. In this case, the distance between (1,4) and (5,2) is greater than the distance between (1,4) and (3,3) we replace **q** with point (5,2).

JARVIS MARCH (GIFT WRAPPING)

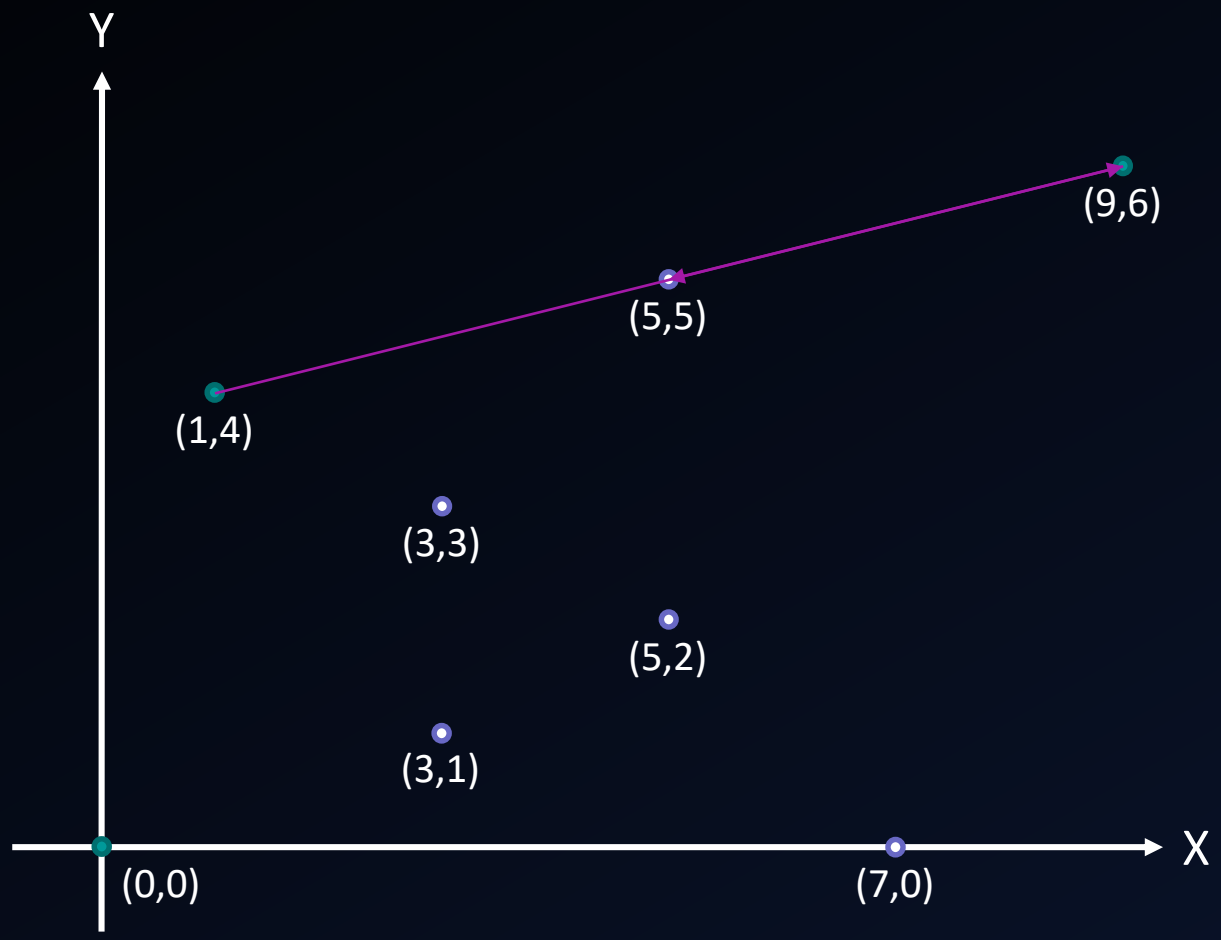
An example. STEP 3.4. The green color means the point is in the convex hull.



Let **j**=(5,5). The sequence [(1,4),(5,5),(5,2)] turns right. We replace **q** with point (5,5).

Jarvis march (Gift wrapping)

An example. STEP 3.5. The green color means the point is in the convex hull.



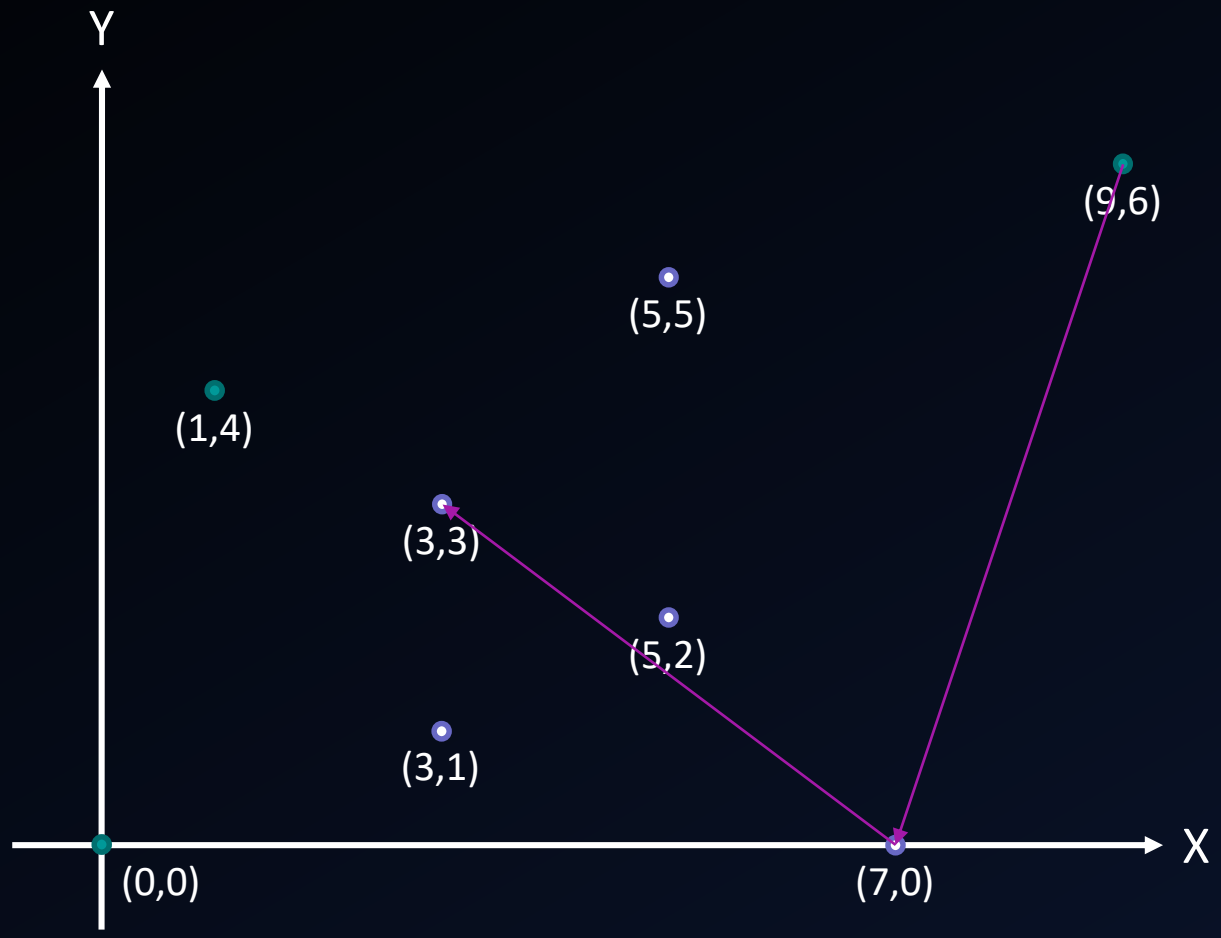
Finally, the last choice for **j** is (9,6). The sequence [(1,4),(9,6),(5,5)] is collinear. In the case of collinear, we replace **q** with **j** only if the distance between **p** and **j** is greater than the distance between q and **p**. In this case, the distance between (1,4) and (9,6) is greater than the distance between (1,4) and (5,5) we replace **q** with point (9,6).

We went through all the points and now **q**=(9,6) is the leftmost point.

We add point (9,6) to the convex hull.

Jarvis march (Gift wrapping)

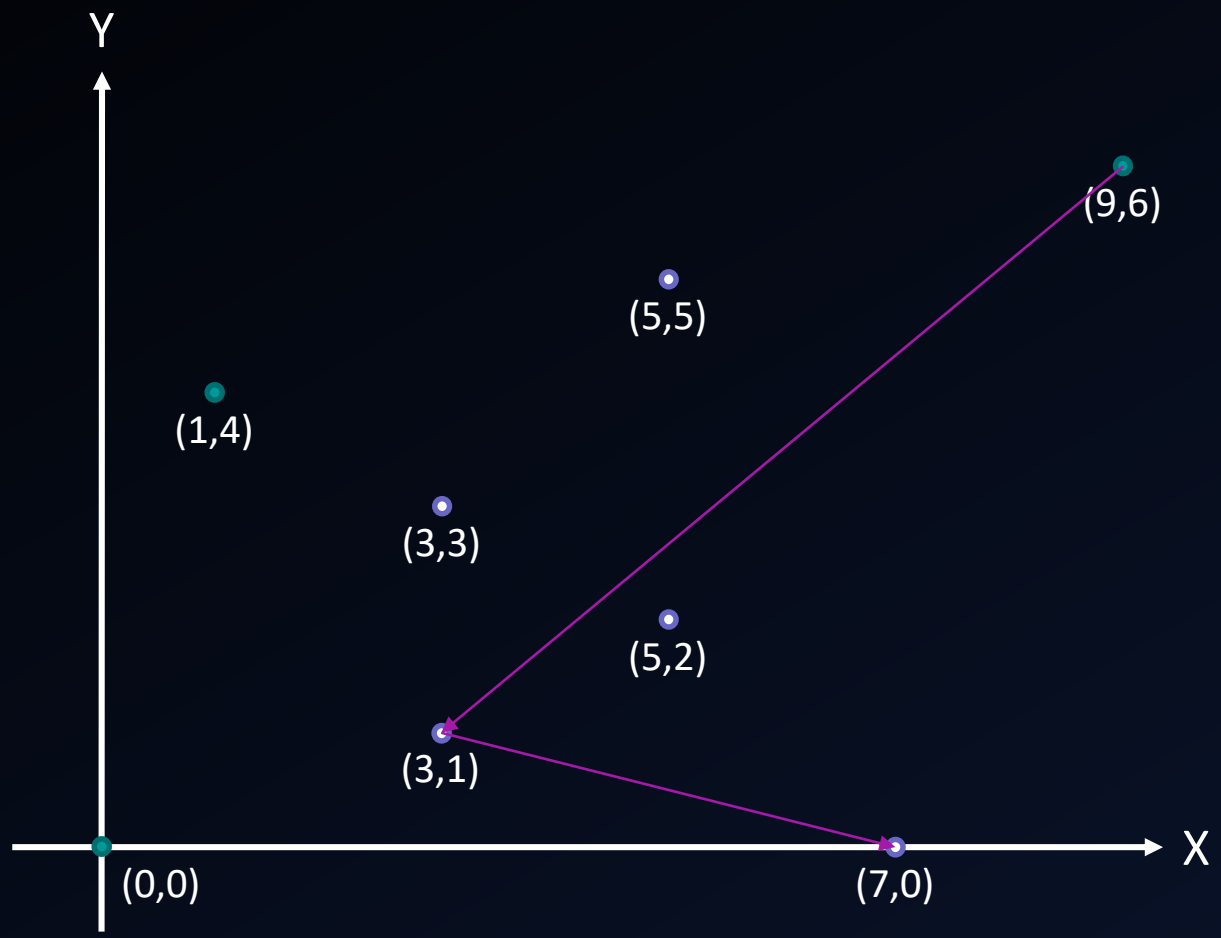
An example. STEP 4.1. The green color means the point is in the convex hull. The previously added points checking are not shown during this step.



Next, we find the leftmost point from the point (9,6) to the convex hull. Let **q** be the point (3,3). Let **j**=(7,0). The sequence [(9,6),(7,0),(3,3)] turns right. We replace **q** with point (7,0).

JARVIS MARCH (GIFT WRAPPING)

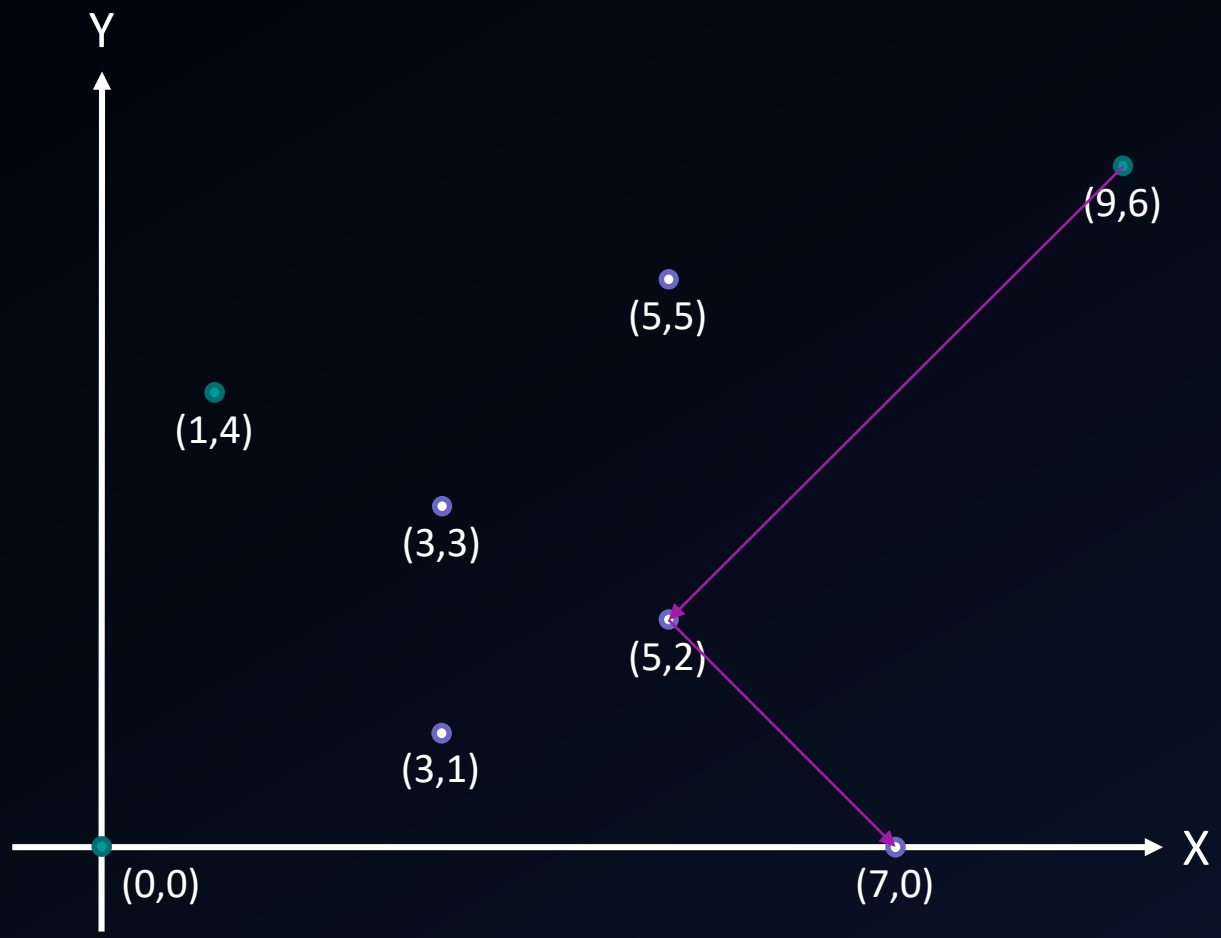
An example. STEP 4.2. The green color means the point is in the convex hull.



Let **j**=(3,1). The sequence [(9,6),(3,1),(7,0)] turns left and we move on without doing anything.

JARVIS MARCH (GIFT WRAPPING)

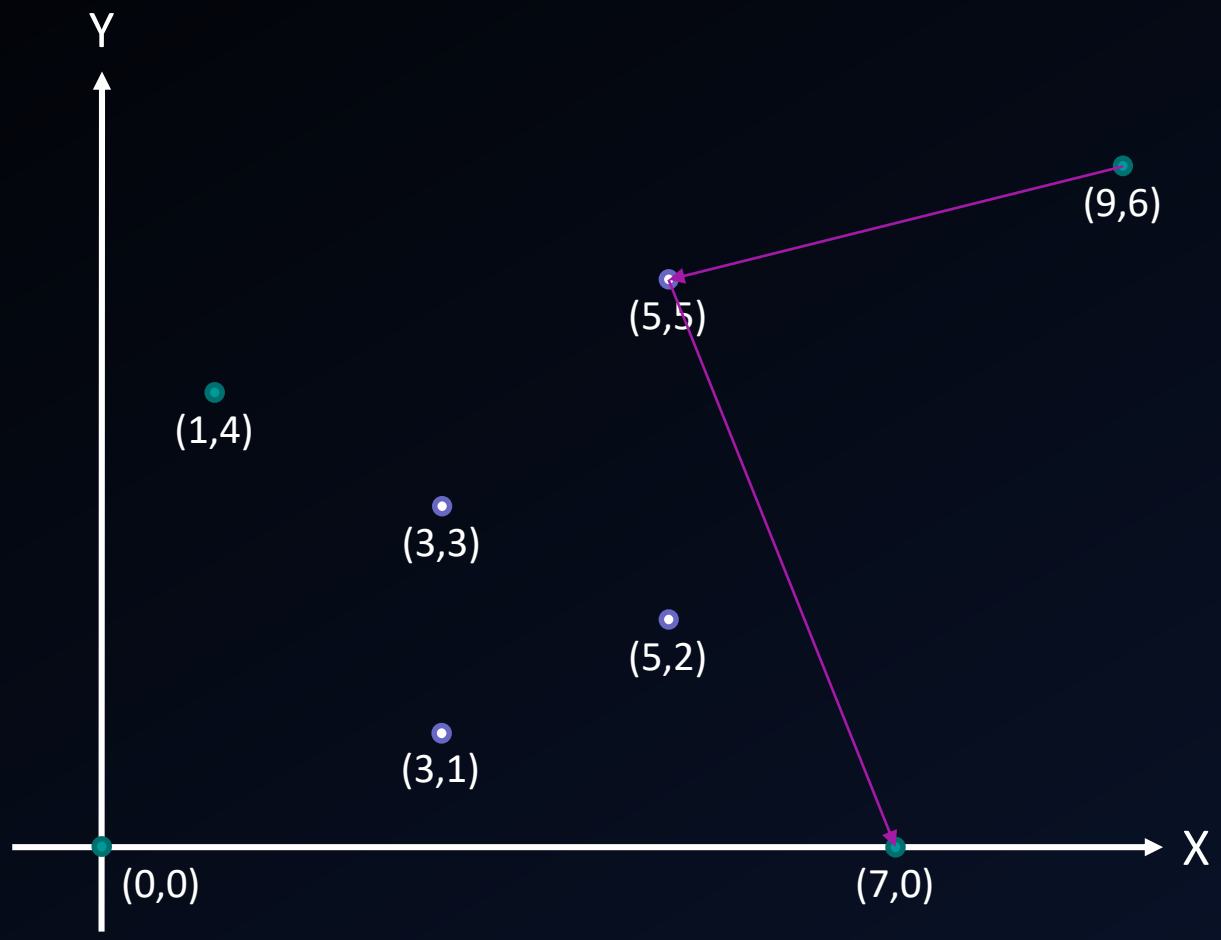
An example. STEP 4.3. The green color means the point is in the convex hull.



Let **j**=(5,2). The sequence [(9,6),(5,2),(7,0)] turns left and we move on without doing anything.

Jarvis march (Gift wrapping)

An example. STEP 4.4. The green color means the point is in the convex hull.



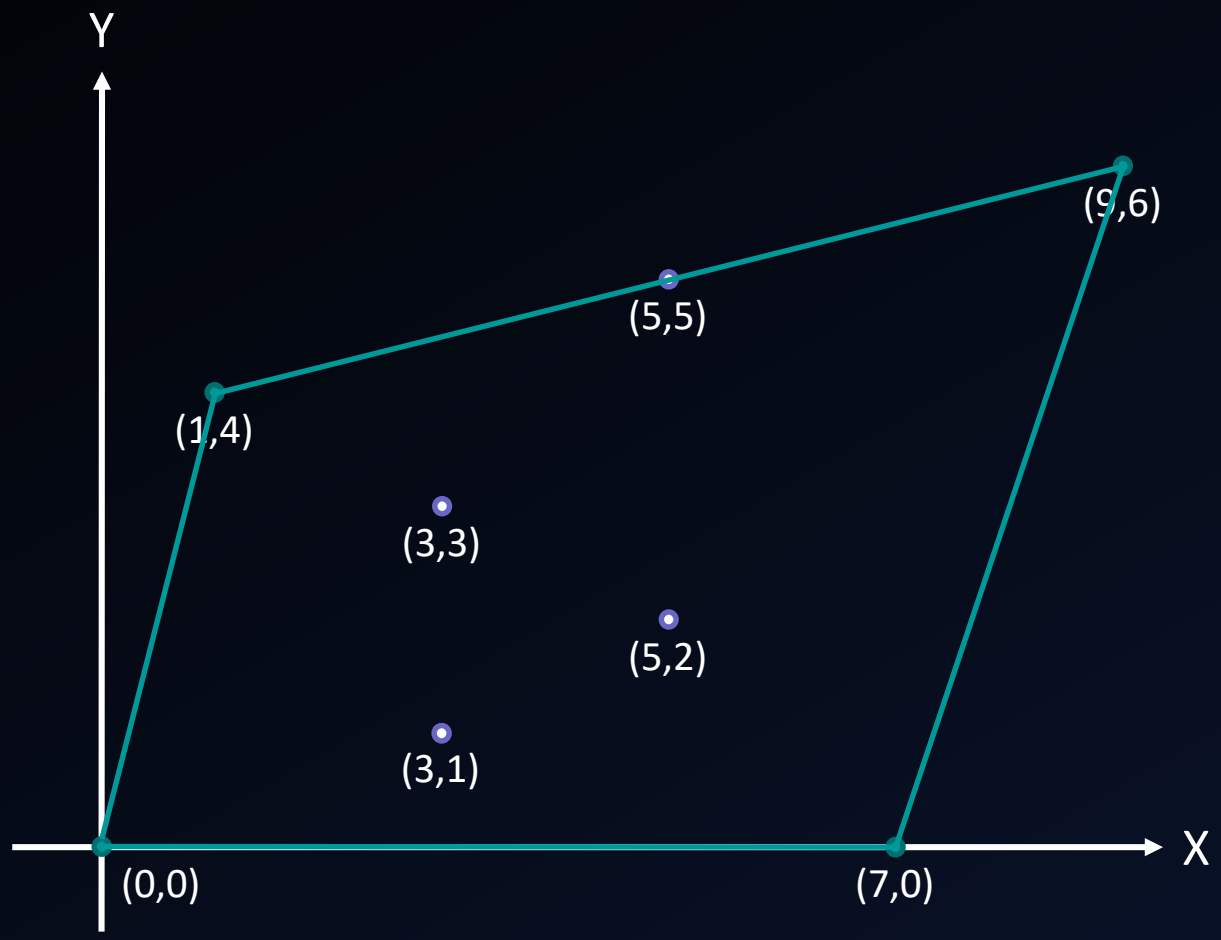
Finally, the last choice for **j** is (5,5). The sequence [(9,6),(5,5),(7,0)] turns left and we move on without doing anything.

We went through all the points and now **q**=(7,0) is the leftmost point.

We add point (7,0) to the convex hull.

Jarvis march (Gift wrapping)

An example. STEP 5. The green color means the point is in the convex hull.



Finally, from (7,0) we compute the leftmost point. The leftmost point from (7,0) will be the point (0,0). Since (0,0) is already in the convex hull, the algorithm stops.

Thank you!