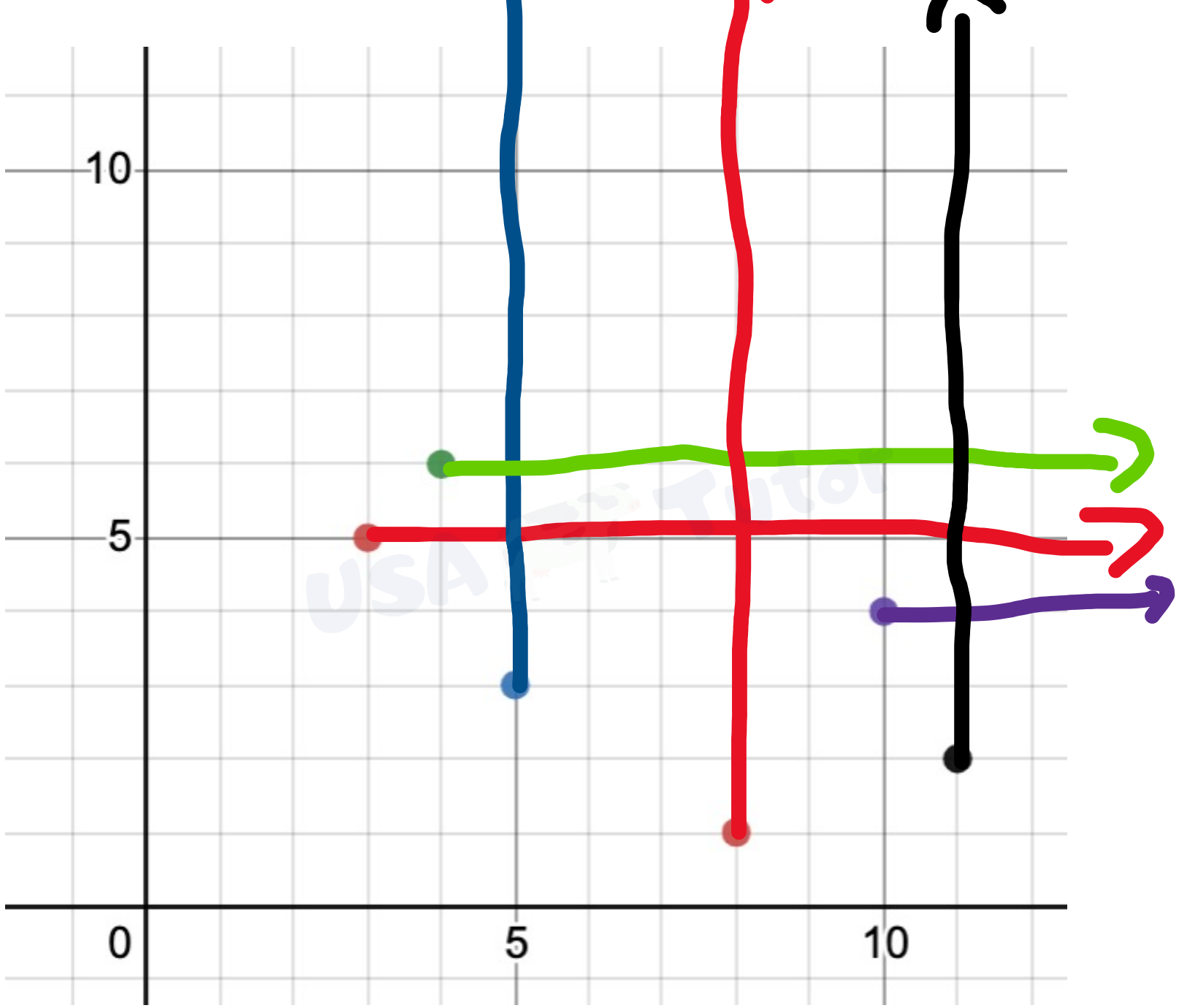
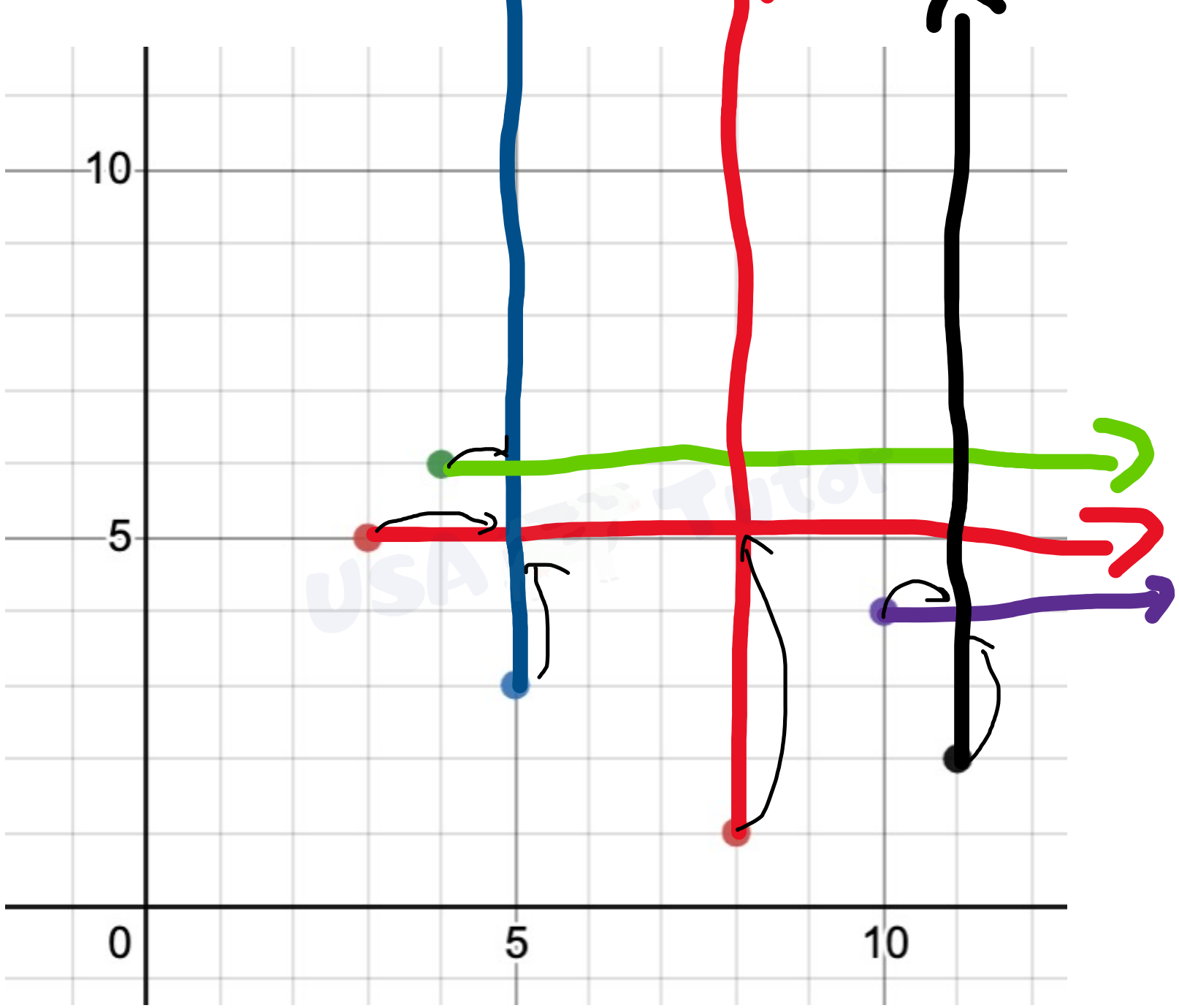
USA Tutor

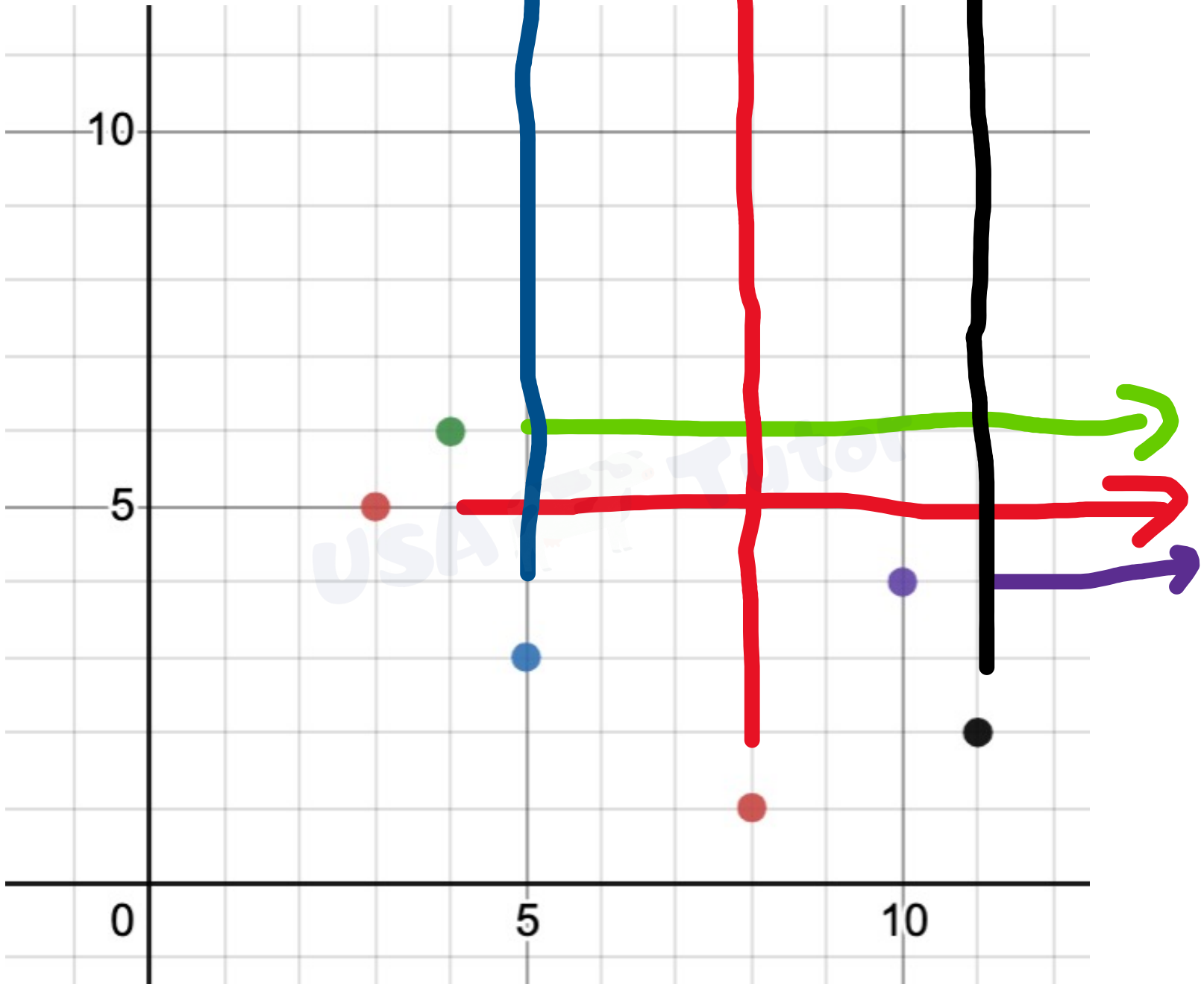
Stuck in a Rut

Analysis by David Yang

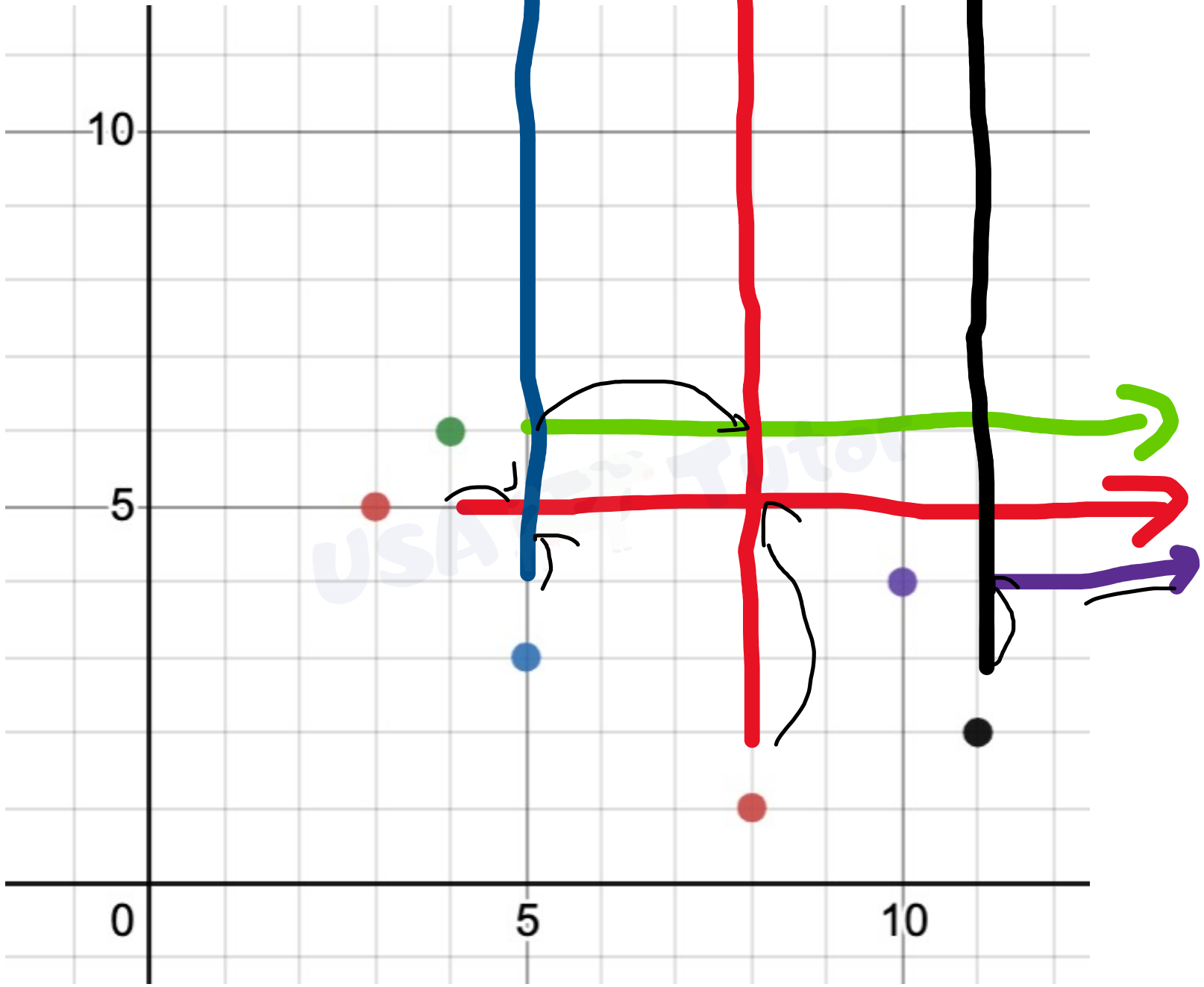

10
5
0
5
10


10
5
0
5
10

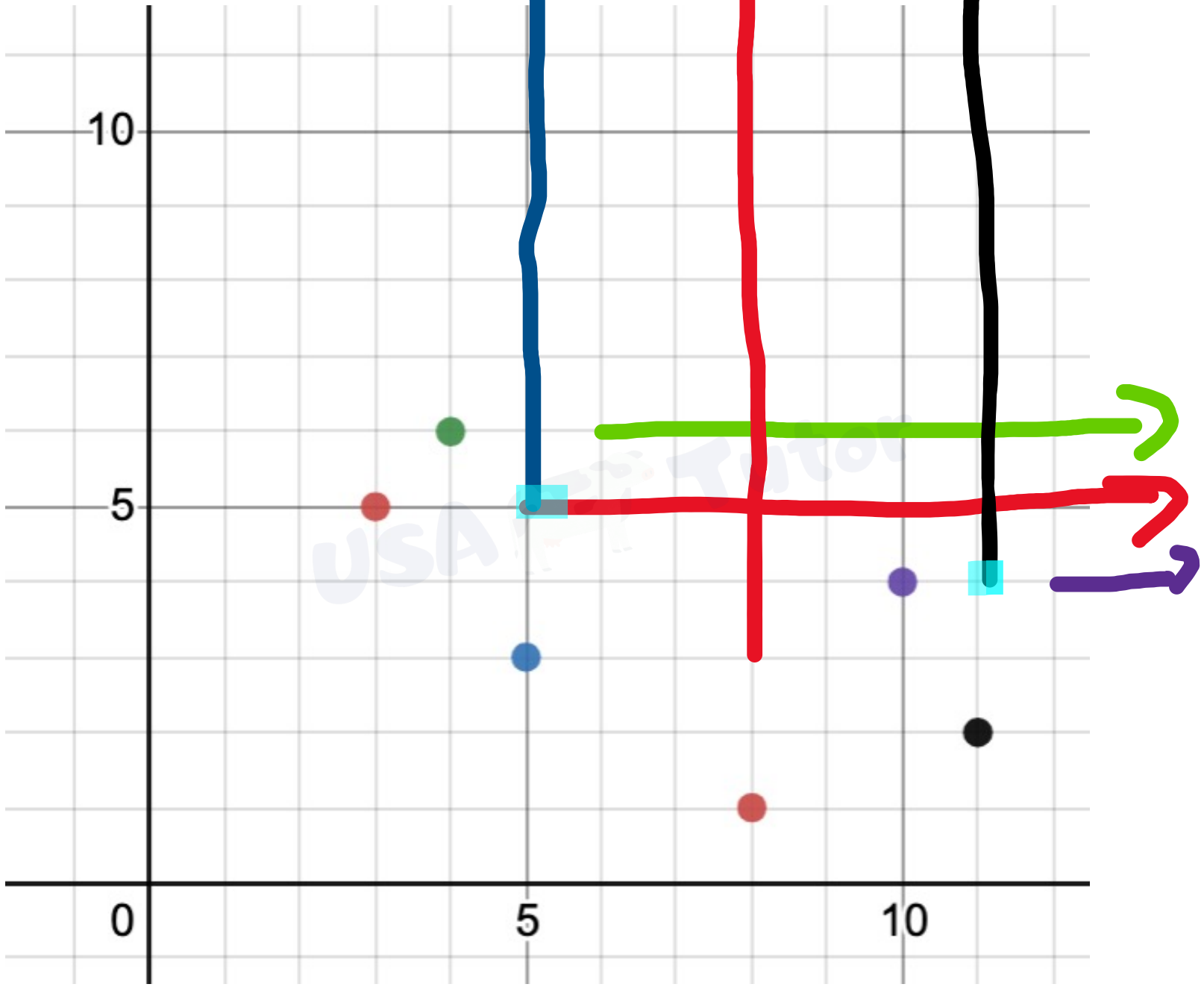
$m_{1,2}=1$



$m_{IA=I}$

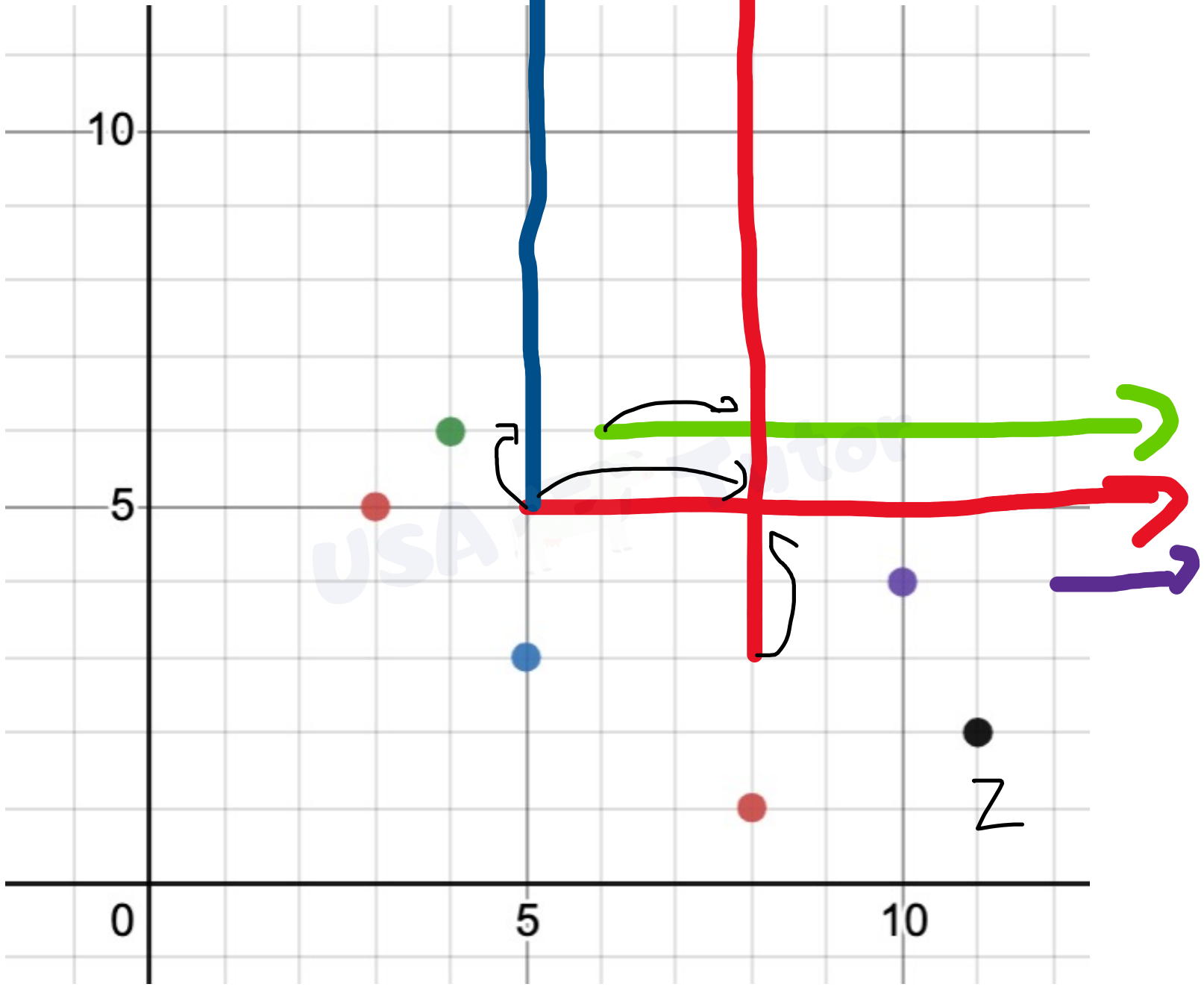


$m_{AB} = 1$

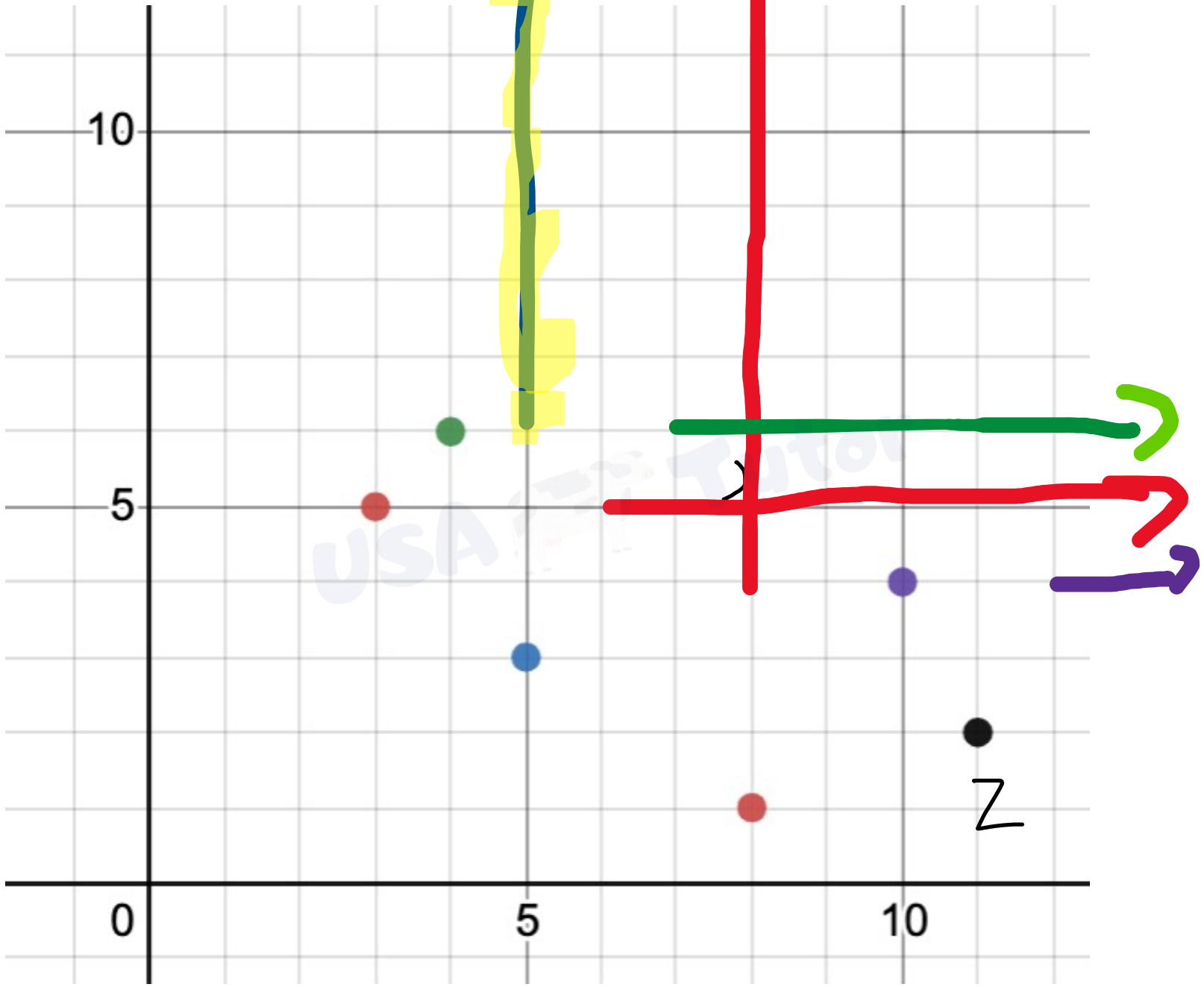


$m_{AB} = 1$

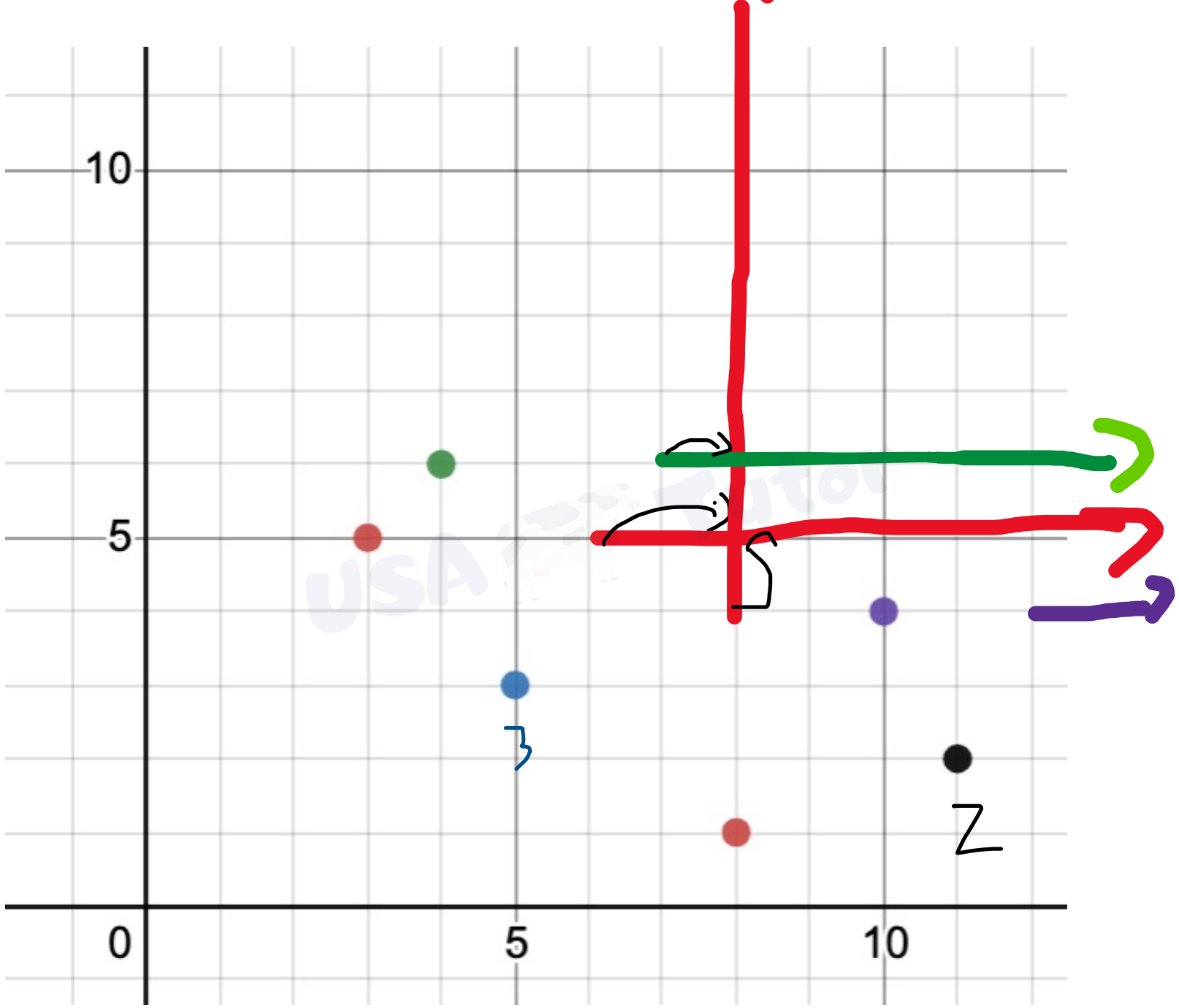
$m_{AB} = 1$



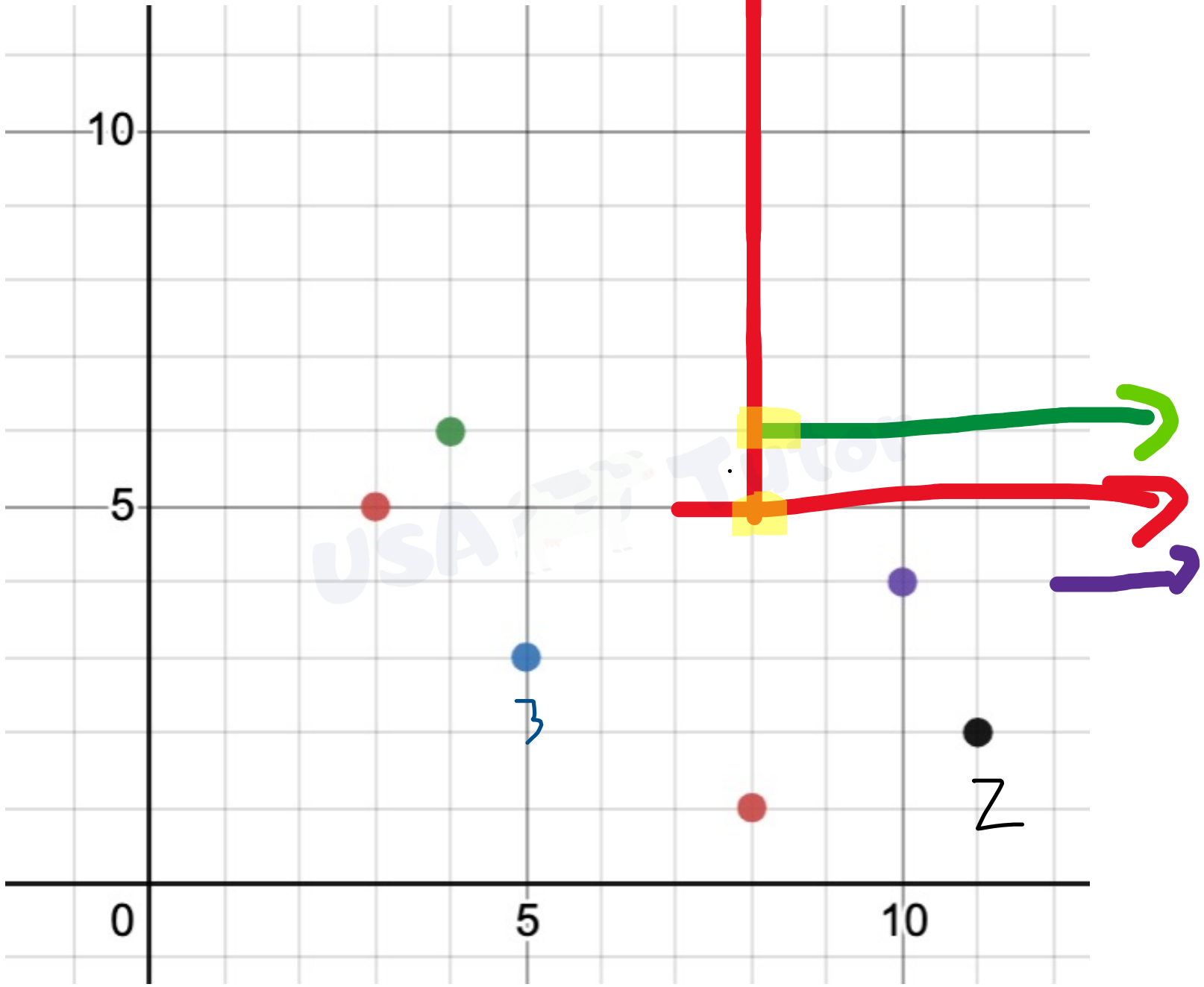
$m_{AB} = 1$



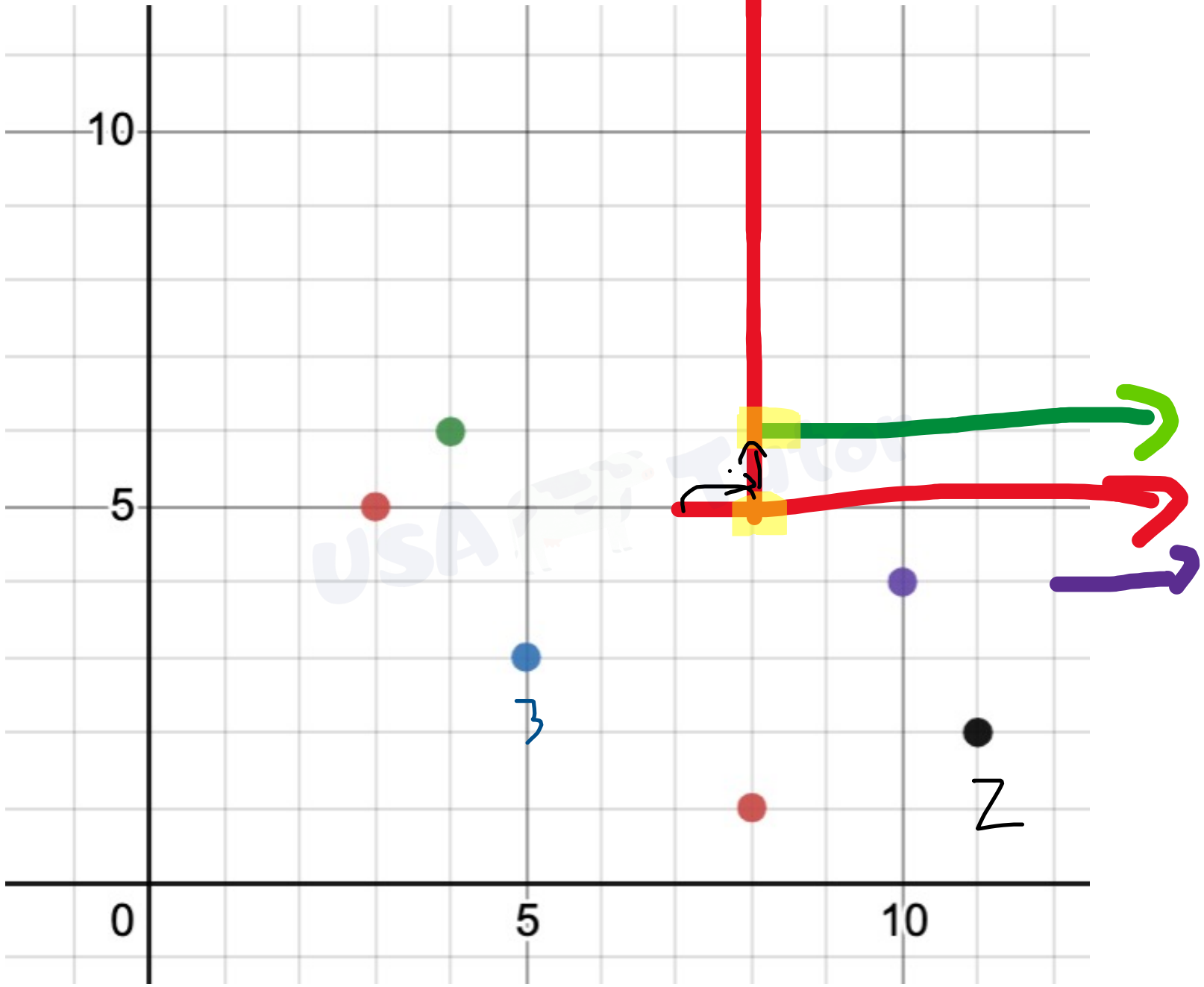
$m_{AB} = 1$



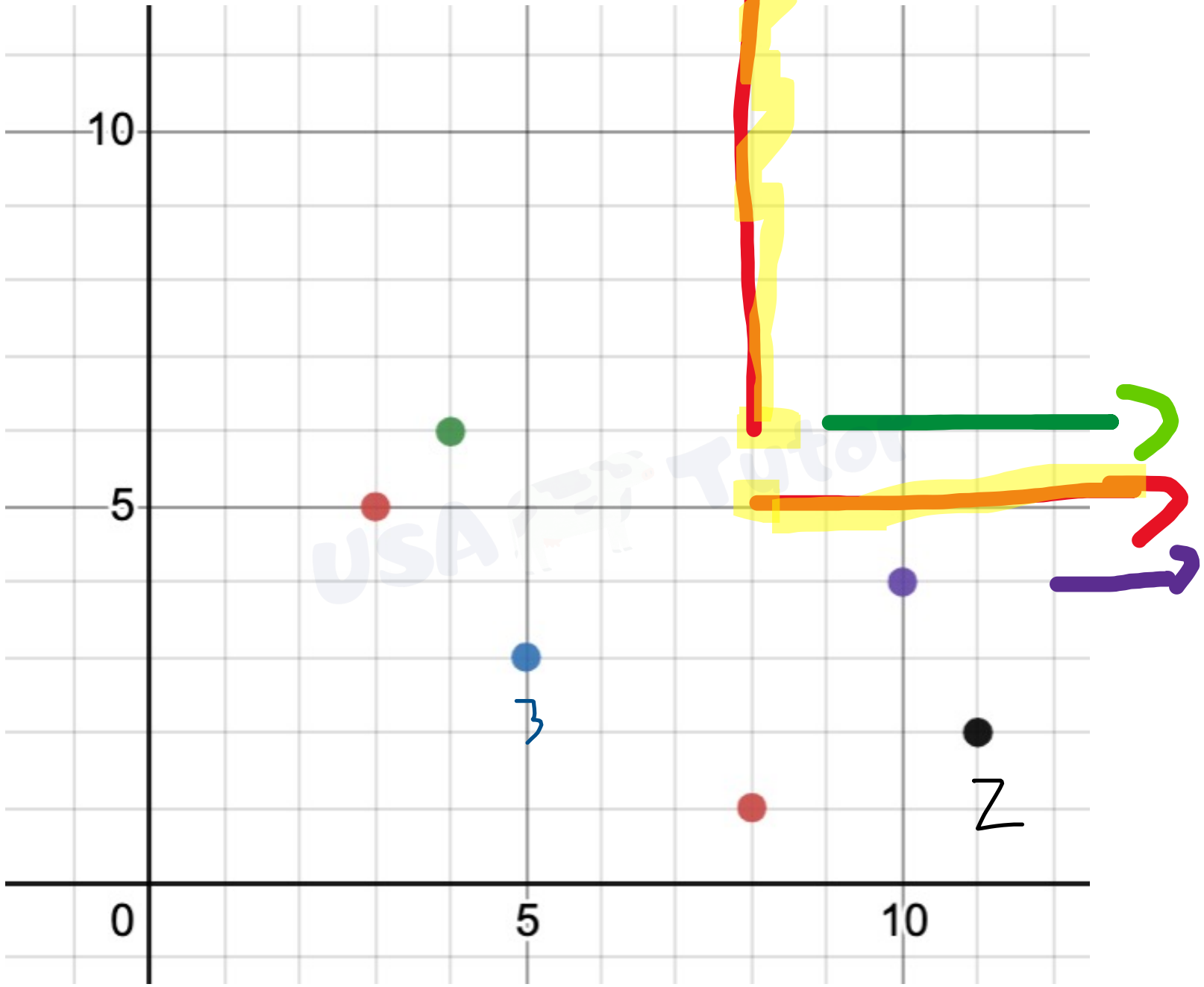
$m_{AB} = 1$



$m_{AB} = 1$

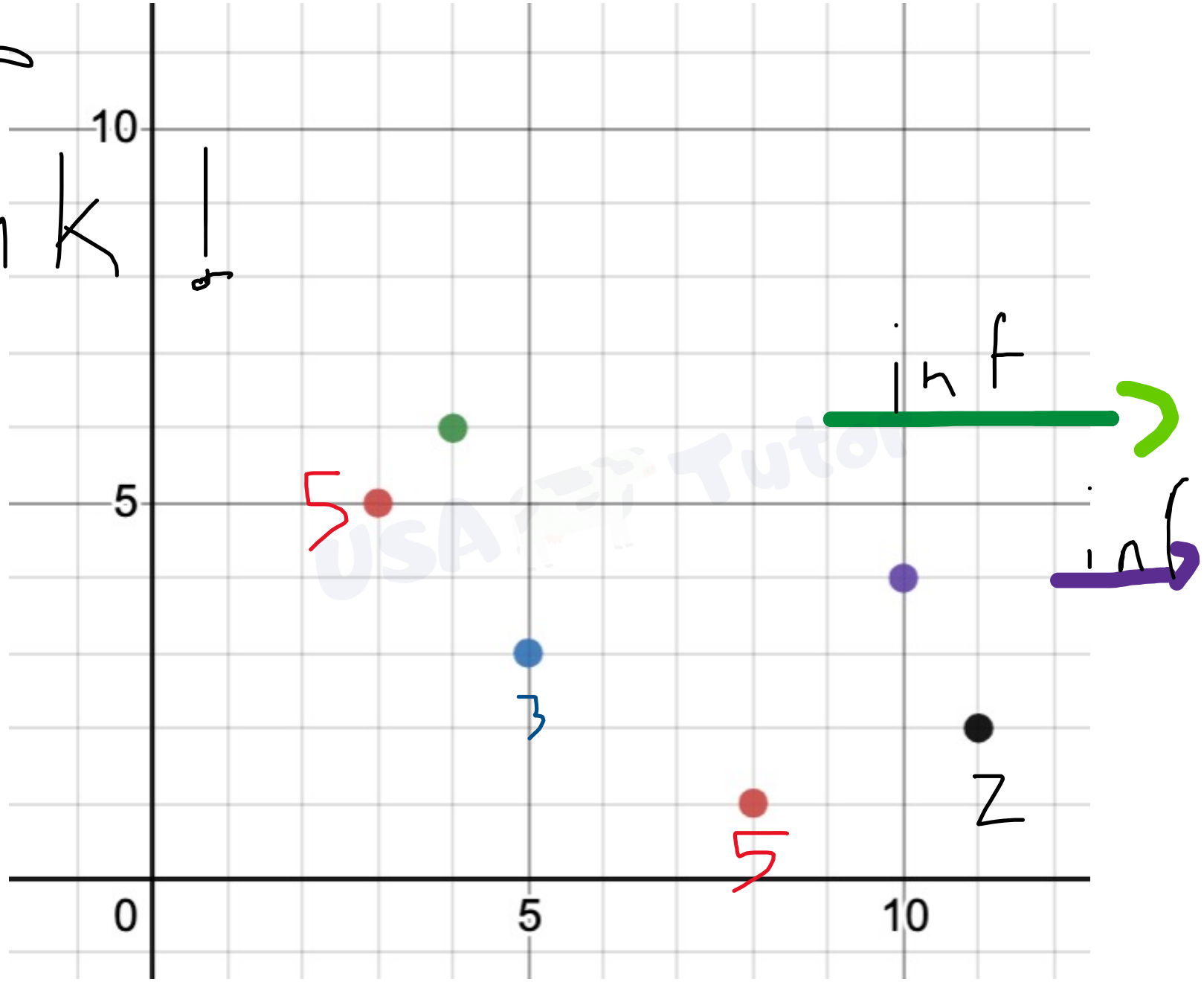


$m_{AB} = 1$



min = −∞

break



Code Sketch

3 Pieces:

1. Finding the Minimum Cost every time
2. Moving all of the cows
3. Marking spots as visited, and stopping “stuck in a rut” cows

Part 1

Finding the Minimum Cost every time:

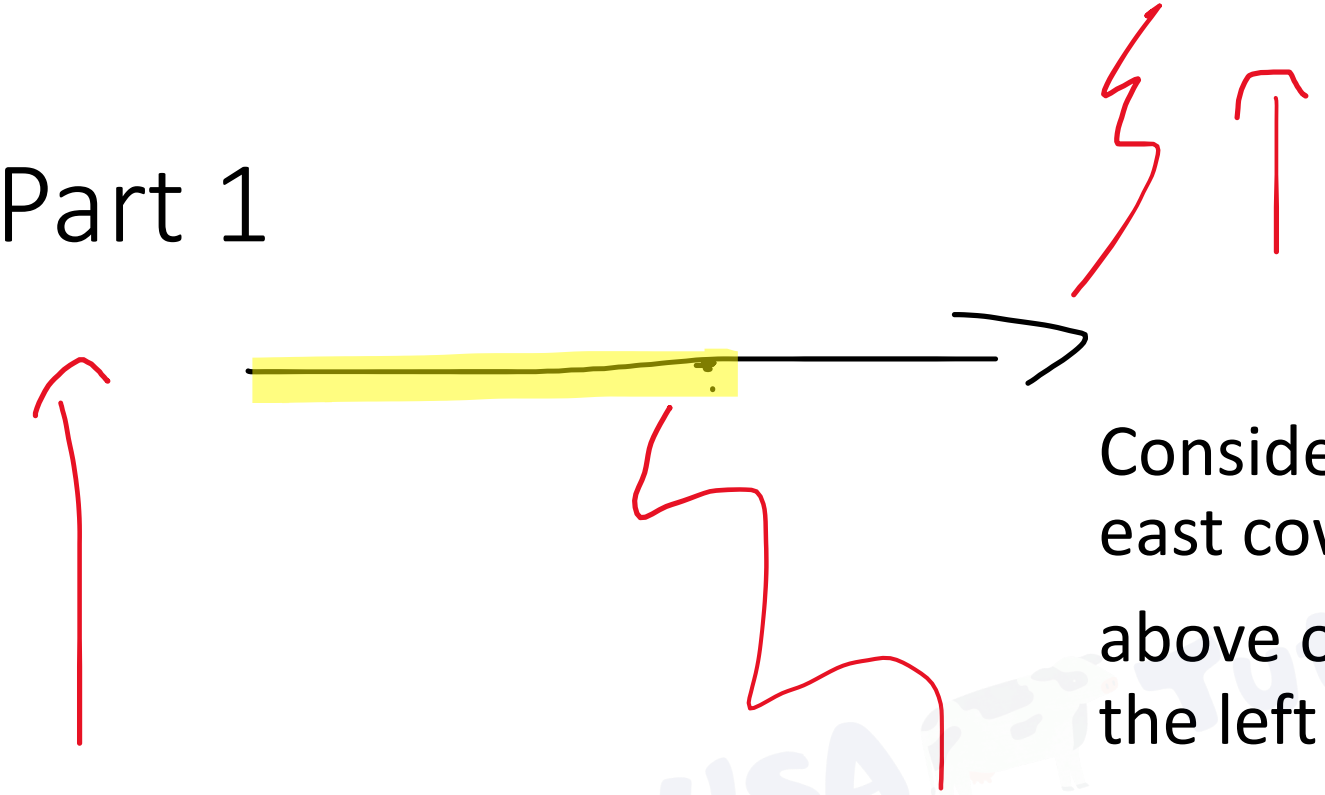
Iterate through every single cow:

For each cow:

Find how long it would take to intersect

// Realize that only north cows can intersect with east, and vice versa

Part 1



Considering we are at a North Cow, our east cow must be

above of us and the starting point to the left of us.

Considering we are at an East Cow, our north cow must be to the right of us and the starting point to the below of us.

Part 2

2. Moving all of the cows

Stuff to do:

If its infinity, break

Mark stuff as visited with a hashset (we cannot use an array because our bounds are hugeeeeee

I use the form x+“:” +y

Part 3

3. Stopping “stuck in a rut” cows

Stuff to do:

Deal with the “come at the same time” case

Create a “stuck” set such that it stores the indexes of stuck cows

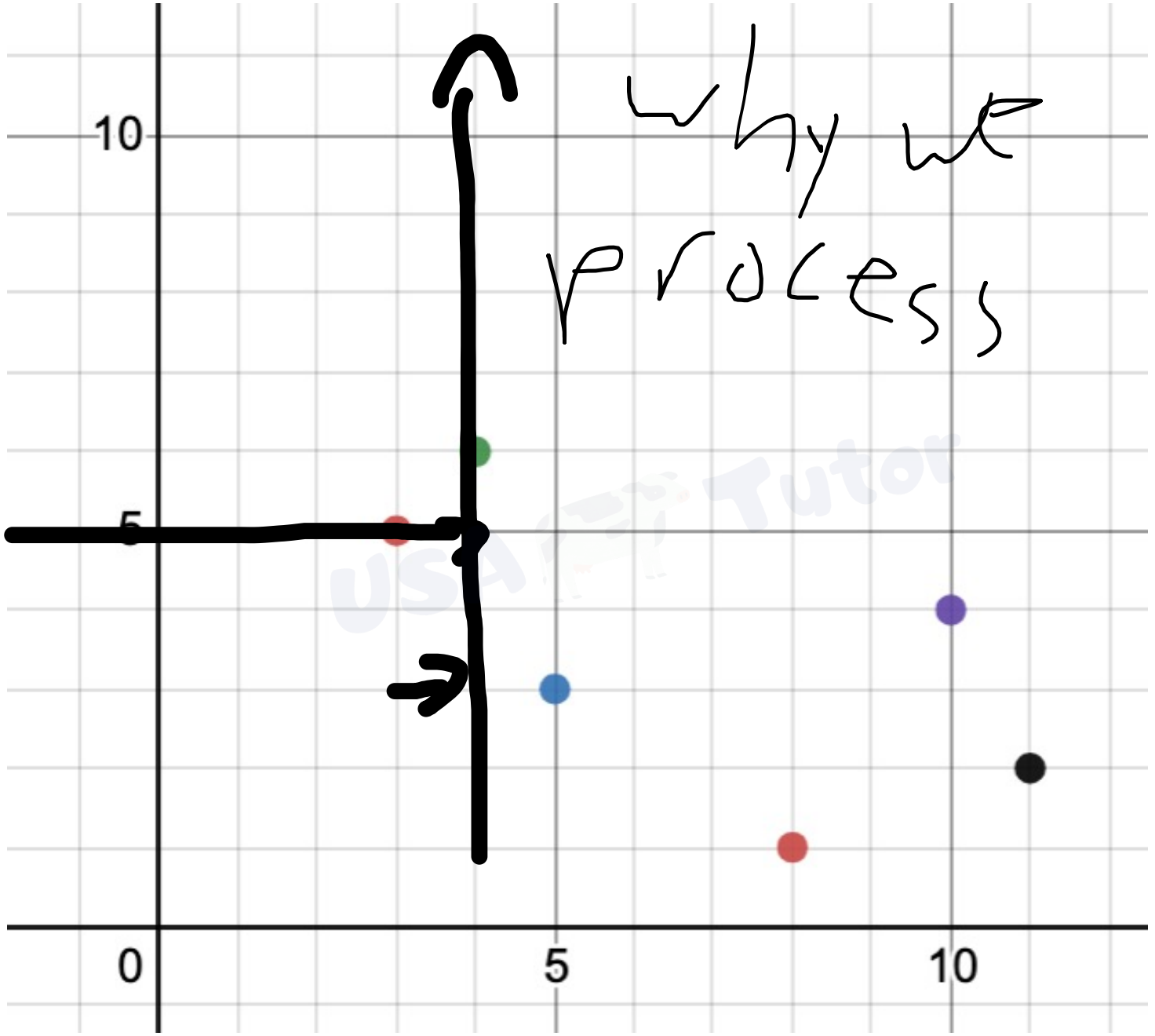
Add a cow to the set, iff we dealt with the prev case, and it’s visited.

Actually “STOP” the cows that are stuck

Solution

If cow i is stuck, then we want to print out how much it moved.

Otherwise, if its not stuck… it’s going infinity.


10
5
0
5
10
why we
process