**Equivalence:**i. point(1,0,1) and point (2,0,2) are same

According to the formula

p×q=[p(y)q(z)−p(z)q(y),p(z)q(x)−p(x)q(z),p(x)q(y)−p(y)q(x)]

They are the same point

Therefore, product=[0\*2-1\*0,1\*2-1\*2,1\*0-0\*1]=[0,0,0]

ii.(1,1,0) and (1,0,1) are not the same point

According to formula

p\*q=[1\*1-0\*0,0\*1-1\*1,1\*0-1\*1]=[1,-1,-1] doesn’t equal to [0,0,0]

They are not the same point

To confirm whether these two points lie on the same line

Suppose the equation of the line is ax+by+cz=0

Since 1\*(-1) +1\*(1) + 0\*(1)=0

1\*(-1) + 0\*(1) + 1\*(1)=0

Therefore both of two points(1,1,0) and (1,0,1) lie on the line (-1,1,,1)

**Recipe of generate\_all\_points(mod):**

Input: mod(the given modulus)

output: uniquepoints(a sequence of unique points each repsented as a tuple of three elements)

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1.Create an empty sequence named uniquepoints, which is used to store all the generated points

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2.Create three variables number0, number1, number2 as three elements of a point represented as a tuple.

Iterate every number (first element of point)and assign the value to the variable number0 over set [0,mod)

Inside the iteration of the variable number0, iterate all numbers (the second element) of point and assign their values to the variable number1 over set [0,mod)

Inside the iteration of the variable number1, iterate all numbers (the third element) of point and assign their values to variable number2 over set [0,mod)

Inside the iteration of the third element of point----the variable number2, determine if number0 or number1 or number 2 equals to zero

If not all of three elements number0, number1, number2 equal to zero(which means number0=0 or number1=0 or numnber2=0),create a tuple named totalpoints , which consists of three elements number0,number1,number2

Add the tuple (number0,number1,number2) into the tuple.

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3. Iterate every number over set [0,length of the sequence uniquepoints) and assign the value to the variable position1

Find the length of the sequence uniquepoints

Inside the iteration of position1, iterate every number over set[position1+1,(length of the sequence uniquepoints)-1) and assign the value to the variable position2

Inside the interation of position2, determine whether the point, with the index of position1 in the sequence uniquepoints equivalent to the point, with the index of position2 under a given modulus

by calling the function equivalent() with uniquepoints[position1], uniquepoints[position2] as its arguments

If two points equal to each other, remove the point with the index of position2

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4.Return the sequence uniquepoints as a list of points each represented as a tuple of three elements

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Recipe of create\_cards(points, lines, mod)

Inputs: points(the given points reach represented as a tuple of three elements), lines(the given lines reach represented as a tuple of three elements), mod(given modulus)

Outputs: a sequence named total\_cards which is a list consisted of a list of integers

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1.Create a new empty sequence named total\_cards, which is used to store all cards

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2.Create an empty seuqence named total\_cards, which is used to store cards

Iterate all cards(represented as tuples) over the given lines, then create an empty sequence named new\_lines, which will be later used to store pictures on the cards

Inside the iteration, iterate all picture(representng images on the card)over given points

Inside the second iteration, determine whether each image belongs to each card by calling the function incident() with picture, cards, mod as arguments

If the image(the variable picture) belongs to each card(the variable cards), transform all images(named pictures consisting of a sequence of tuples that have three elements each ) to integers.

Then push all these integers onto the sequence new\_lines

Push the the sequence new\_lines that contains cards(with intergers that represent images on the card) onto the empty sequence total\_cards

Return the sequence total\_cards

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Discussion

It is not possible.

Because we can first think about a image in one random card, supposing you have n images per card.

The maximum amount of the image we can find in cards is n-1 because one card is held by us.

There are in total of n images, each appear (n-1) times, so we need n\*(n-1) more cards.

However, we also need to add the initial card that is held by us. So we should have n\*(n-1)+1 cards

in total. Also the number of images and cards must be integers.

However if we want a valid deck of 40 cards, we must let n\*(n-1)+1=40.

In this way, n(the number of images per card) will not be an integer.

So it is not possible.