Equivalence

i: (1,0,4) crossed with (2,0,8) = (0\*(8) - 4\*(0), 4\*(2) - 1\*(8), 1\*(0) – 0\*(2)) = (0,0,0)

ii: (0,0,1) crossed with (1,0,0) = (0(0)-1(0), 1(1) – 0(0), 0(0)-0(1)) = (0,1,0)

(0,0,1) lies on {0,1,0} since (0\*0+0\*1+1\*0) = 0

(1,0,0) lies on {0,1,0} since (1\*0+0\*1+0\*0) = 0

The first function is a called generate\_all\_points which will receive one input named mod which is an integer. With that input, it will create all the possible unique points given the mod. It does this by first creating an empty sequence which will hold all possible points assign to the variable points\_list. Next, we iterate over from 0 to the integer value of mod. Within that iteration we iterate again from 0 to the integer value of mod. Then we iterate within that inner iteration from 0 to the integer value of the mod. Then we create an ordered triple(point) from the current number iteration we are in from all three iterations and we add that to our sequence named points\_list. After it has gone through all those iterations we remove the ordered triple (0,0,0) from points\_list because it is not a valid point in spot it. Then we create a new sequence named unique\_list which will hold all the possible unique points. The next step is we iterate through every point in points\_list. At every iteration, we check if the current point we are on is equivalent to any of the points in unique\_list using the helper function equivalent. If the current point was not equivalent to any of the points in unique\_list. Then it gets added to the sequence unique\_list. At the end of all the iterations we now have all the unique points for the given mod and the function will return unique\_list.

The next function is called create\_card which will receive three inputs named points and lines which are sequences filled with ordered triples and mod which is just an integer. With those inputs, the function will create a valid deck for the game spot it. It does this by first creating a sequence that will hold all the cards named list\_of\_cards. Then we iterate through all the lines and within every iteration we create an empty sequence that will hold all the items named card. Then we iterate through all the points and check if it’s on the current line using the function incident. If it is then we add the integer index value that the point represents to the sequence card. After we have added all the points that are on the current line we add the sequence card to the sequence list\_of\_cards. After all the iteration have happened then have all the cards function will return list\_of\_cards.

Discussion: It is possible to create a valid deck of 40 “Spot it!” cards. Since if we create a valid deck of 57 cards using the modulus 7. Then we can remove 17 of those cards and the deck should remain valid. The reasoning is since before we removed any cards, every card should only have one match with every other card. This wouldn’t have changed by us removing any cards. Every card still has only one match with every other card.