Secret Sharing

1. **Situation&Intuition** :

   (a) You have a secret that is so powerful that you don't want one person to be able to get to it alone, but instead you want least k people to come together to be able to get to it.

   (b) In order to make sure at least k people come together, and any number of people less than k would not be able to access the secret, we use polynomials and their property that d+1 points on a degree d polynomial uniquely defines the polynomial. Any number less than d+1 points would not allow to determine the polynomial with certainty.

   (c) The idea is to hide the secret in a polynomial of correct degree based on wanted k, and then distribute points from the polynomial to people. If the cutoff for being able to uniquely get the correct polynomial P(x) is k people, and each people corresponds to a point, then P(x) should be a degree k-1 polynomial. A degree (k-1) polynomial need k distinct points to uniquely create P(x), if each person has one point, then at least k people has to work together, use their points to get P(x).

   (d) So, k people (or more) together can combine their points and get the unique P(x) polynomial through Lagrange interpolation. Notice on the other hand that any number of people less than k wouldn't have enough points on the degree (k-1) polynomial to uniquely interpolate P(x).

   (e) Once they get P(x) through interpolation, they just plug in P(0) (or wherever you decided to hide the secret on the polynomial) to get the secret s.

2. **Set – up** : **Howdoyouhideyoursecret**?

   (a) First, set your secret s to be y-intercept of the polynomial: P(0) = s (y-intercept is arbitrary, technically you can encode your secret s to anywhere you want in the polynomial as long as people know where it is too so they can recover it at the end)

   (b) Assuming you want at least k people to combine together to access the secret, P(x) should be a (k-1) degree polynomial.

   (c) To create the specific (k-1) degree polynomial with P(0)=s you want to use, pick k points for your polynomial to cross. One of the points obviously has to be P(0) = s. But for the other points, you can randomly pick any pair of x and y coordinates for your P(x) to cross.

   (d) For example if k = 3, your P(x) is degree 2, and you need 3 points to define your own P(x). We know one of the points is P(0) = 2. The remaining 2 points can be randomly chosen such as P(1) = 15, P(79) = 2, P(-4) = 23, since you will build the polynomial P(x) based off these points. But for easy calculation, I'd suggest just going with assigning manageable values to P(1), P(2).

   (e) Once you pick k points, use lagrange interpolation to get your very own P(x). It crosses through your secret s, and the other points you picked. Yay! Now that you have the polynomial P(x), you can crease as many points on the polynomial as you want as P(1), P(2), ... P(1000), P(1001) ... and distribute these points to people as you like.

   (f) Notice that because all these points that you produce are coming from your P(x) any distinct k of them together can be used to interpolate the polynomial P(x) and access the secret, but less than k points wouldn't be enough to uniquely determine P(x) since it is a k-1 degree polynomial.

3. **Example&Explanation : DIS3C, Q1**

    (a) n countries with k representatives each in the UN. The vault can be opened if (i) all n countries come together or (ii) at least m countries come together with SG.

    (b) Firstly, start with the crowded case of all countries. Clearly we can't make countries share a point so each country gets their own point. If we want to open the vault with n countries, that means n points should be able to recover the polynomial. So polynomial is of degree n-1. Now case (i) is satisfied, all countries together can open the vault. Once you know how many points you need in total, it is easier to distribute it to fewer people like in case (ii). Now we know m¡n countries have m points in total, and since they need a total of n points to access the vault, Secretary General must provide the difference: she has n-m distinct points. So m countries and SG has n points as well.

    (c) To add extra security, UN wants to make sure all of the k representatives of the country to agree before country takes any action. Whenever there's a case of extra security, consider ways for countries to use their points more difficult. In order to make sure all k representatives agree to use the country-point, we can create country-specific polynomials, give a point from that polynomial to all of the country's representatives, and select its degree such that only if all k representatives came together with their points, they could access the country-point and use it in the UN. Since all k reps must agree, we want a polynomial uniquely describable with k points, which means it should be of degree (k-1). So now, before a country can help, they have to combine all their reps to solve the country-polynomial and get the country-point which is the secret hidden in the country-specific polynomial.

4. **Tips&Warnings**:

    (a) It is important that the points you distribute to people are distinct (different x values), using the same point twice in interpolation doesn't give new info to access P(x). So if you give the same point to two people and they end up coming together to access the secret, they only have one distinct point.

    (b) Make sure not to distribute P(0) as one of the points since that's your secret.

    (c) While trying to satisfy multiple conditions, start with the crowded case. Remember, you can give multiple points to a person but you can't give one point to multiple people if there's a chance they might want to work together. Starting from the crowded case helps you select correct degree of your polynomial.

Error Correction

1.