Create a github repository with the all the info.

LETS start with a single case or an easy case.

Read the data.

1. Create a small data set to practice.
   1. With small samples to practice the algorithm.
2. Read the data and put it on data frames.
3. Do not need to change the ASCCI because it will be the same and we will lose time doing the change.

Start algorithm

1P: ABD

2P: ABD

Frame: ABD

Frame: A

3

2

1. The frames will need to have the number of tags as well it will have.
2. Join all the portraits to create the frames.
   1. Join frames that have most things different.
   2. Find a way to know how to find the most different portraits to join.
      1. <https://math.stackexchange.com/questions/1774801/minimizing-the-intersection-of-three-sets>
      2. minimize the intersection of two sets
      3. <https://stackoverflow.com/questions/62845996/reducing-a-list-of-sets-to-minimum-by-intersecting-them>
   3. Join all landscapes and while this orders them.
3. We must try to have ordered all the frames of portraits in here so that we can save time.
4. Once we have the portraits put the landscapes in frames.
5. Sort all the frames considering the number of tags each one has and ordered from the one that has more to the one that has less.
   1. Search for a sorting algorithm that work good with half of an array sorted.
   2. <https://www.geeksforgeeks.org/sorting-algorithms/>
6. Organize the output data.

* Is there a time requirement? An hour
* What do we have to return? 4 outputs
* How many tags do we have?
* How many executions we have to do?

Idea do an extra sort between the ones that have the same amount of tags to see which has the better combination of elements.

Sort all the tags of a painting in alphabetical order.

And have a list of all the tags that can exist and that way we create an array.

Count the number of tags that start with a letter and have an array of number of letters each one have and compare it with another.