

Estudiantes:

- Wyo Hann Chu – 202015066 – Req 3
- Thomas Gomez – 202112869 – Req 4

Requerimiento 1:

```
def listarcronologicamente(catalog,ano_inicial,ano_final):
    artists=catalog['artists']
    ms.sort(artists,cmpfunction1)
    for indice in range(1,lt.size(artists)+1):
        if int(lt.getElement(artists,indice)['BeginDate'])==ano_final and in
t(lt.getElement(artists,indice+1)['BeginDate'])>ano_final:
            indice_final=indice
        if int(lt.getElement(artists,indice)['BeginDate'])==ano_inicial and
int(lt.getElement(artists,indice-1)['BeginDate'])<ano_inicial:
            indice_inicial=indice
        sizesublist=indice_final-indice_inicial
        listadeseada=lt.subList(artists,indice_inicial,sizesublist)
```

Como podemos ver en el peor de los casos el algoritmo necesita recorrer toda la lista de artista, por lo tanto, el orden de crecimiento es lineal, en otras palabras,  $O(n)$ , donde  $n$  es numero de datos de artists

Requerimiento 2:

```
artwork_1 = lt.getElement(ord_artworks,int(pos_1))
```

```
id_artwork1 = str(artwork_1['ConstituentID']).replace("[", "").replace("]", "")
```

```
list_id_artwork_1 = id_artwork1.split(",")
```

```
artistas_artwork1 = ""
```

```
for x in list_id_artwork_1:
```

```

for y in range(1,lt.size(artist)+1):

    if str((lt.getElement(artist,y))['ConstituentID']).replace("[", "").replace("]", "") == x:
        artistas_artwork1 = str((lt.getElement(artist,y))['DisplayName'])
        break

print('Titulo: ' + artwork_1['Title'] + ' Artista(s): ' + str(artistas_artwork1) + ' Date Acquired: ' +
      artwork_1['DateAcquired'] + ' Medio: ' + artwork_1['Medium'] + ' Dimensions: ' +
      artwork_1['Dimensions'])

artwork_2 = lt.getElement(ord_artworks,int(pos_1+1))
id_artwork2 = str(artwork_2['ConstituentID']).replace("[", "").replace("]", "")
list_id_artwork_2 = id_artwork2.split(",")
artistas_artwork2 = ""

for x in list_id_artwork_2:
    for y in range(1,lt.size(artist)+1):

        if str((lt.getElement(artist,y))['ConstituentID']).replace("[", "").replace("]", "") == x:
            artistas_artwork2 = str((lt.getElement(artist,y))['DisplayName'])
            break

print('Titulo: ' + artwork_2['Title'] + ' Artista(s): ' + str(artistas_artwork2) + ' Date Acquired: ' +
      artwork_2['DateAcquired'] + ' Medio: ' + artwork_2['Medium'] + ' Dimensions: ' +
      artwork_2['Dimensions'])

```

```

artwork_3 = lt.getElement(ord_artworks,int(pos_1+2))
id_artwork3 = str(artwork_3['ConstituentID']).replace("[", "").replace("]", "")
list_id_artwork_3 = id_artwork3.split(",")
artistas_artwork3 = ""

for x in list_id_artwork_3:
    for y in range(1,lt.size(artist)+1):

        if str((lt.getElement(artist,y))['ConstituentID']).replace("[", "").replace("]", "") == x:
            artistas_artwork3 = str((lt.getElement(artist,y))['DisplayName'])
            break

print('Titulo: ' + artwork_3['Title'] + ' Artista(s): ' + str(artistas_artwork3) + ' Date Acquired: ' +
      artwork_3['DateAcquired'] + ' Medio: ' + artwork_3['Medium'] + ' Dimensions: ' +
      artwork_3['Dimensions'])

```

```

artwork_4 = lt.getElement(ord_artworks,int(pos_2-2))
id_artwork4 = str(artwork_4['ConstituentID']).replace("[", "").replace("]", "")
list_id_artwork_4 = id_artwork4.split(",")
artistas_artwork4 = ""

for x in list_id_artwork_4:
    for y in range(1,lt.size(artist)+1):

        if str((lt.getElement(artist,y))['ConstituentID']).replace("[", "").replace("]", "") == x:
            artistas_artwork4 = str((lt.getElement(artist,y))['DisplayName'])
            break

print('Titulo: ' + artwork_4['Title'] + ' Artista(s): ' + str(artistas_artwork4) + ' Date Acquired: ' +

```

```
artwork_4['DateAcquired'] + ' Medio : ' + artwork_4['Medium'] + ' Dimensions: ' +  
artwork_4['Dimensions'])
```

```
artwork_5 = lt.getElement(ord_artworks,int(pos_2-1))  
id_artwork5 = str(artwork_5['ConstituentID']).replace("[", "").replace("]", "")  
list_id_artwork_5 = id_artwork5.split(",")  
artistas_artwork5 = ""  
  
for x in list_id_artwork_5:  
    for y in range(1,lt.size(artist)+1):  
        if str((lt.getElement(artist,y))['ConstituentID']).replace("[", "").replace("]", "") == x:  
            artistas_artwork5 = str((lt.getElement(artist,y))['DisplayName'])  
            break  
  
print('Titulo: ' + artwork_5['Title'] + ' Artista(s): ' + str(artistas_artwork5) + ' Date Acquired: ' +  
artwork_5['DateAcquired'] + ' Medio: ' + artwork_5['Medium'] + ' Dimensions: ' +  
artwork_5['Dimensions'])  
  
artwork_6 = lt.getElement(ord_artworks,int(pos_2))  
id_artwork6 = str(artwork_6['ConstituentID']).replace("[", "").replace("]", "")  
list_id_artwork_6 = id_artwork6.split(",")  
artistas_artwork6 = ""  
  
for x in list_id_artwork_6:  
    for y in range(1,lt.size(artist)+1):  
        if str((lt.getElement(artist,y))['ConstituentID']).replace("[", "").replace("]", "") == x:  
            artistas_artwork6 = str((lt.getElement(artist,y))['DisplayName'])  
            break  
  
print('Titulo: ' + artwork_6['Title'] + ' Artista(s): ' + str(artistas_artwork6) + ' Date Acquired: ' +
```

```
artwork_6['DateAcquired'] + ' Medio: ' + artwork_6['Medium'] + ' Dimensions: ' +  
artwork_6['Dimensions'])
```

Encontrar la asignación que más se repite en este algoritmo de manera analítica es de gran dificultad, por ende graficaremos y veremos la tendencia del algoritmo, prediciendo de manera experimental el orden de crecimiento. Como se analiza la lista artworks, utilizamos el tamaño de los datos de esta

Requerimiento 3 “Wyo Chu” :

```
def printSortResultsArtistArtowrk_Teqc(ord_artworkId,codigo_artista):
```

```
    size = lt.size(ord_artworkId)
```

```
    total_obras = 0
```

```
    total_medios = 0
```

```
    tipos_medios = lt.newList()
```

```
    for x in range(1, size+1):
```

```
        if codigo_artista in  
str(lt.getElement(ord_artworkId,x)['ConstituentID']).replace("[", "").replace("]", ""):
```

```
            total_obras += 1
```

```
            if lt.isPresent(tipos_medios, lt.getElement(ord_artworkId,x)['Medium']) == 0:
```

```
                lt.addLast(tipos_medios, ( lt.getElement(ord_artworkId,x)['Medium'], 1))
```

```
                total_medios += 1
```

```
            else:
```

```
                pos = lt.isPresent(tipos_medios, lt.getElement(ord_artworkId,x)['Medium'])
```

```
                element = lt.getElement(tipos_medios, pos)
```

```
                lt.changeInfo(tipos_medios,pos,(element[1],element[2]+1))
```

```

print("total obras", str(total_obras))
print("total medios", str(total_medios))

num_medio_mas_usado = 0
medio_mas_usado = ""
print('Medios usados y su cantidad:')
for x in range(1, lt.size(tipos_medios)+1):
    print( lt.getElement(tipos_medios,x))
    if lt.getElement(tipos_medios,x)[1] > num_medio_mas_usado:
        num_medio_mas_usado = lt.getElement(tipos_medios,x)[1]
        medio_mas_usado = lt.getElement(tipos_medios,x)[0]

print('El medio mas usado es: ', str(medio_mas_usado))
print("obras con el medio mas usado")
for x in range(1, size+1):
    if codigo_artista in
str(lt.getElement(ord_artworkId,x)['ConstituentID']).replace("[", "").replace("]", ""):
    if medio_mas_usado in str(lt.getElement(ord_artworkId,x)['Medium']):
        print('Titulo: ', str(lt.getElement(ord_artworkId,x)['Title']), ' Fecha:
',str(lt.getElement(ord_artworkId,x)['Date']),' Medio:
',str(lt.getElement(ord_artworkId,x)['Medium']),'Dimensiones',
str(lt.getElement(ord_artworkId,x)['Dimensions']))

```

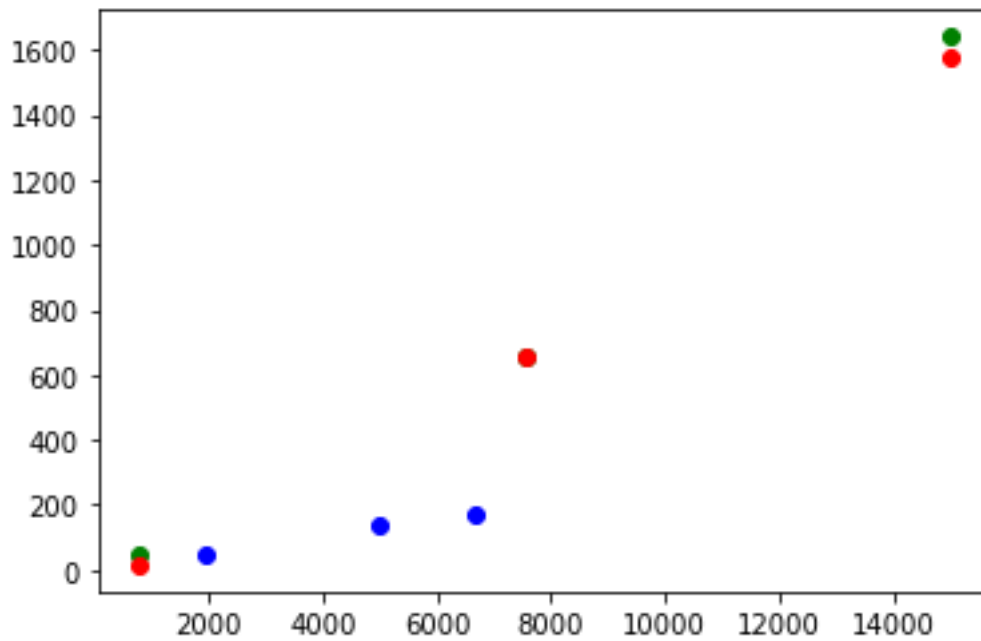
Encontrar la asignación que más se repite en este algoritmo de manera analítica es de gran dificultad, por ende graficaremos y veremos la tendencia del algoritmo, prediciendo de manera experimental el orden de crecimiento, como el requerimiento depende del tamaño de artworks, utilizamos el tamaño de estos.

Graficas:

#blue reque 1

#green reque 2

#red requie 3



Como podemos el requerimiento 1(puntos azules) crece más lento y se ajusta mejor a una ecuación lineal.

Los requerimientos 2 y 3 (puntos verdes y rojos) tiendes a ajustarse menos a una curva lineal, estos mismos tienden a crecer polinomial-mente, siendo el grado del polinomio mayor o igual a 2:

Podemos concluir que el algoritmo mas eficiente es el requerimiento 1