Universidad de los Andes
Facultad de Ingeniería
Reto 1- Curando y Explorando el MoMA (Modern Museum of Art)
Estructuras de datos y algoritmos – sección 02

Estudiante 1:

Nombre: Sofia Torres Ramírez.
Código: 2020140872
Correo uniandes: s.torres21@uniandes.edu.co

Estudiante 2:

Nombre: Ana Margarita Flórez Ruiz.
Código: 201922242
Correo uniandes: a.florezr@uniandes.edu.co

Análisis de complejidad de cada requerimiento

REQ. 1: Listar cronológicamente los artistas (Grupal)

```
# REQ. 1: listar cronológicamente los artistas

def addartistyear(catalog, año1, año2):
    artistsinrange=lt.newList("ARRAY_LIST")
    i=1
    while i<= lt.size(catalog["Artista"]):
        artista=lt.getElement(catalog["Artista"],i)
        if int(artista["BeginDate"])>= año1 and int(artista["BeginDate"])<=año2:
            lt.addLast(artistsinrange, artista)
        i+=1
    sortedlist=sortyear(artistsinrange)
    return sortedlist

# Funciones de ordenamiento
def sortyear (artistsinrange):
        sorted_list=mg.sort(artistsinrange, cmpArtworkByBeginDate)
        return sorted_list

# Funciones utilizadas para comparar elementos dentro de una lista
def cmpArtworkByBeginDate (date1, date2):
    if date1["BeginDate"]!= "" and date2["BeginDate"]!= "":
        year1= int((date1["BeginDate"]))
        year2= int((date2["BeginDate"]))
        return year1<year2
```

Complejidad: O(n)

Este requerimiento cuenta con una complejidad de O(n) debido a que como se puede ver en el código, se cuenta con un while que debe recorrer los n elementos de la lista dada “catalog” para así poder encontrar los artistas que estén dentro del rango de años dados por parámetro.

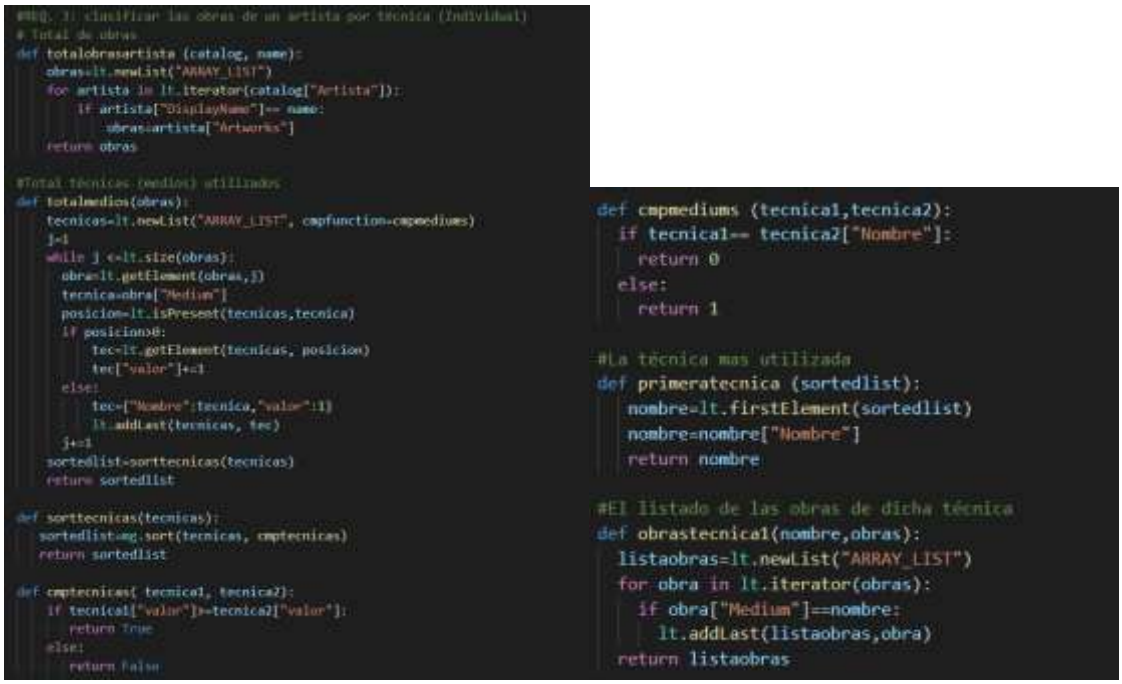
REQ. 2: Listar cronológicamente las adquisiciones (Grupal)

Complejidad: O(n)

En este requerimiento cuenta con una complejidad de O(n) ya que, como en el requerimiento anterior, se tienen iteraciones dentro de los datos de tipo “while”. Como se puede ver, existen dos ciclos en este requerimiento, uno en la función “addartworkyear” el cual retorna una lista ordenada considerando las dos fechas ingresadas, esta función tiene una complejidad de O(n). En la función “purchaseart” también se hace uso de un ciclo para poder encontrar la cantidad de obras que fueron compradas y la complejidad de esta también es O(n). A pesar de que el requerimiento cuenta con dos funciones de complejidad O(n), al sumar estas dos complejidades se obtiene O(2n), esta complejidad seguirá siendo de O(n).

REQ. 3: Clasificar las obras de un artista por técnica (Individual)

Realizado por Sofía Torres.



```
#REQ. 3: Clasificar las obras de un artista por técnica (Individual)
# Total de obras
def totalobrasartista (catalog, name):
    obras=lt.newList("ARRAY_LIST")
    for artista in lt.iterator(catalog["Artista"]):
        if artista["DisplayName"]== name:
            obras=artista["Artworks"]
    return obras

#Total técnicas (medios) utilizadas
def totalmedios(obras):
    tecnicas=lt.newList("ARRAY_LIST", cmpfunction=cmpmediums)
    j=1
    while j <=lt.size(obras):
      obra=lt.getElement(obras,j)
      tecnica=obra["Medium"]
      posicion=lt.isPresent(tecnicas,tecnica)
      if posicion>0:
          tec=lt.getElement(tecnicas, posicion)
          tec["valor"]+=1
      else:
          tec={"Nombre":tecnica,"valor":1}
          lt.addLast(tecnicas, tec)
      j+=1
    sortedlist=sorttecnicas(tecnicas)
    return sortedlist

def sorttecnicas(tecnicas):
    sortedlist=sg.sort(tecnicas, cmptecnicas)
    return sortedlist

def cmptecnicas( tecnica1, tecnica2):
    if tecnica1["valor"]>tecnica2["valor"]:
        return True
    else:
        return False
```

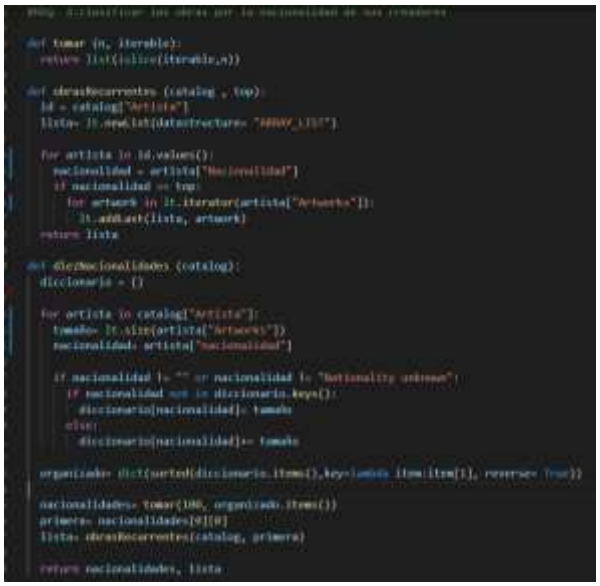
```
def cmpmediums (tecnica1,tecnica2):
  if tecnica1== tecnica2["Nombre"]:
    return 0
  else:
    return 1

#La técnica mas utilizada
def primeratecnica (sortedlist):
    nombre=lt.firstElement(sortedlist)
    nombre=nombre["Nombre"]
    return nombre

#El listado de las obras de dicha técnica
def obrastecnica1(nombre,obras):
  listaobras=lt.newList("ARRAY_LIST")
  for obra in lt.iterator(obras):
    if obra["Medium"]==nombre:
      lt.addLast(listaobras,obra)
  return listaobras
```

Complejidad: O(n)

Este requerimiento tiene una complejidad de O(n), debido a que, al igual que en requerimiento pasado, se tienen varias funciones con complejidad O(n), en este caso serían las funciones “totalobrasartista”, “totalmedios” y “obrastecnica” los cuales recorren todos los datos debido a que hacen uso de un ciclo, sin embargo, al hacer la suma de las complejidades de estas tres funciones se obtiene una complejidad de O(3n), pero se sabe que esta complejidad seguirá siendo O(n).

REQ. 4: Clasificar las obras por la nacionalidad de sus creadores (Individual)

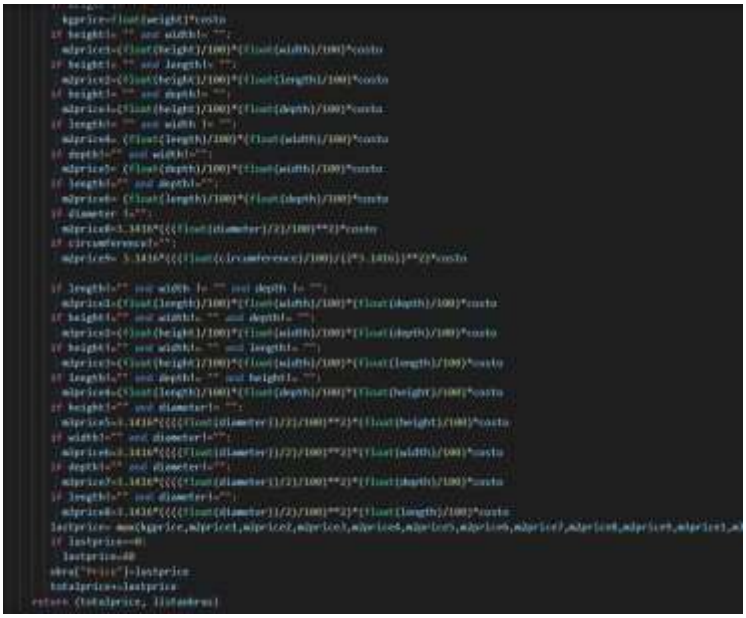
Realizado por Margarita Flórez.



Complejidad: O(n^2)

En cuanto al requerimiento 4, como se puede ver en el código mostrado, es de O(n^2) esto se debe a que hay un punto en el código en el que hay dos iteraciones de tipo for una dentro de la otra, esto provocando un doble recorrido dentro del catalogo para poder clasificar las obras por nacionalidad y de paso poder ordenarlas en su top 10 de manera correcta.

REQ. 5: transportar obras de un departamento (Grupal)



Complejidad: O(n)

En este requerimiento se tiene una complejidad de O(n) debido a que, al igual que el requerimiento 2 y 3, a pesar de que varias funciones hacen uso de ciclos, nunca se hace un ciclo de ciclos por lo que la complejidad de cada una de las funciones con ciclos del requerimiento es de O(n) y como ya se sabe, al trabajar con Big O, la complejidad del requerimiento seguirá siento O (n).

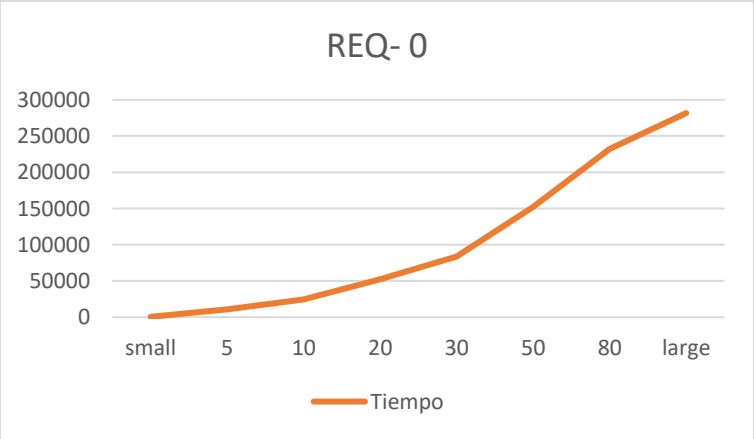
Pruebas de tiempos de ejecución.

Las pruebas de tiempos de ejecución se realizaron es un equipo con las siguientes características:

Características	Máquina
Procesadores	Intel(R) Core (TM) i7-10510U CPU @ 1.80GHz 2.30 GHz
Memoria RAM (GB)	16,0 GB
Sistema Operativo	Sistema operativo de 64 bits, procesador basado en x64

REQ. 0: Carga de datos.

Tamaño del archivo	Tiempo (milisegundos)
Small	500
5	10687.5
10	24343.75
20	52500
30	83859.38
50	152281.25
80	232312.5
large	281593.75

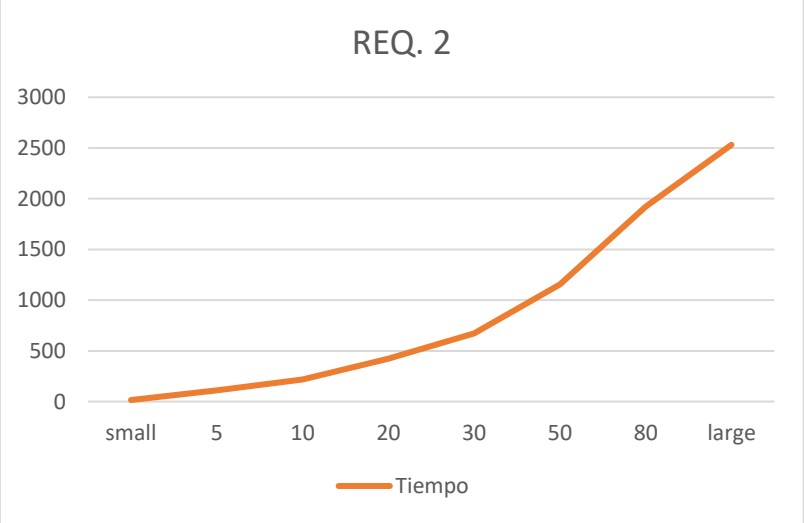


REQ. 1: Listar cronológicamente los artistas (Grupal)

Tamaño del archivo	Tiempo (milisegundos)
Small	15.63
5	78.13
10	109.38
20	140.63
30	156.25
50	187.5
80	234.38
large	234.38

REQ. 2: Listar cronológicamente las adquisiciones (Grupal)

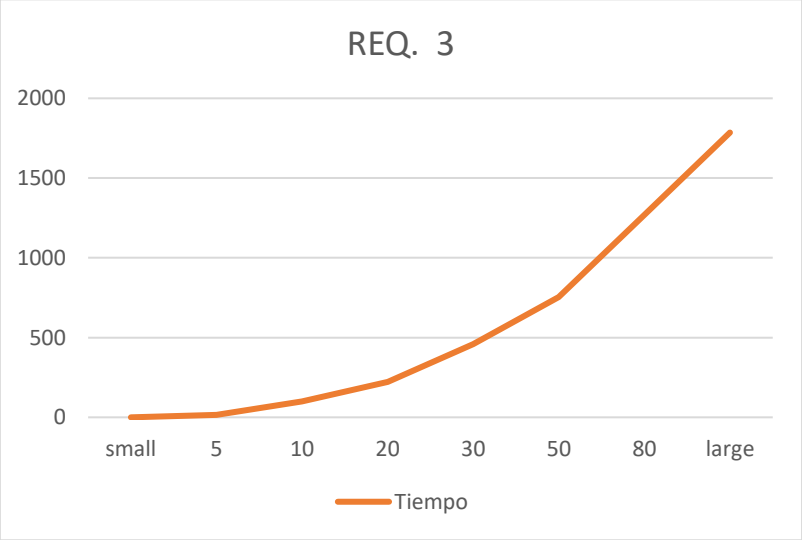
Tamaño del archivo	Tiempo (milisegundos).
Small	15.63
5	109.38
10	218.75
20	421.88
30	671.88
50	1156.25
80	1921.88
large	2531.25



REQ. 3: Clasificar las obras de un artista por técnica (Individual)

Realizado por Sofía Torres.

Tamaño del archivo	Tiempo (milisegundos)
Small	0
5	15.3
10	100.6
20	222.78
30	458.9
50	754.2
80	1269.3
large	1785.4



REQ. 4: Clasificar las obras por la nacionalidad de sus creadores (Individual)

Realizado por Margarita Flórez.

Tamaño del archivo	Tiempo (milisegundos)
Small	16.8
5	156.8
10	321.9
20	439.8
30	671.89
50	987.1
80	1001.6
large	1892.7

REQ. 5: transportar obras de un departamento (Grupal)

Tamaño del archivo	Tiempo (milisegundos)
Small	31.25
5	328.13
10	671.86
20	2234.38
30	2421.88
50	3890.63
80	6171.88
large	7781.25