Clase 10 consulta  
Algoritmos de Búsqueda y ordenación  
  
Búsqueda Lineal  
  
muns=[2,5,4,8,74,15,1]  
  
def busqueda\_lineal(lista, valor):  
for i in range(len(lista)):  
if lista[i] == valor:  
return i  
return -1  
  
a= busqueda\_lineal(muns, 15)  
print(a)  
  
  
busqueda binaria  
  
muns=[2,5,14,18,74,155,160]  
  
def busqueda\_binaria(lista, valor):  
izq = 0  
der = len(lista) - 1  
while izq <= der:  
medio = (izq + der) // 2  
if lista[medio] == valor:  
return medio  
elif lista[medio] < valor:  
izq = medio + 1  
print(izq)  
else:  
der = medio - 1  
print(der)  
return -1   
  
aux=busqueda\_binaria(muns,155)  
print(aux)  
  
  
  
Ejemplo de algoritmo de ordenacion por selección  
  
muns=[2,5,4,8,74,15,1]  
#print(muns)  
def selection\_sort(nums):  
n = len(nums)  
#print(n)  
for i in range(n):  
min\_index = i  
#print(i)  
for j in range(i+1, n):  
  
print(nums[min\_index])  
print(nums[j])   
if nums[j] < nums[min\_index]:  
print("Cambio")   
min\_index = j  
nums[i], nums[min\_index] = nums[min\_index], nums[i]  
return nums  
  
selection\_sort(muns)  
#print(muns)  
  
  
ALGORITMO DE ORDENACIÓN POR INSERCIÓN  
El algoritmo de ordenación por inserción es otro algoritmo simple pero eficiente que  
funciona comparando cada elemento con los elementos que lo preceden y  
colocándolo en su lugar correcto dentro del conjunto. Este proceso se realiza  
iterativamente para cada elemento del conjunto. En este algoritmo, se asume que la  
parte izquierda del conjunto siempre está ordenada y se insertan los elementos de la  
parte derecha en su posición correcta dentro de la parte ordenada.  
  
muns=[2,5,7,8,74,85,91]  
def insertion\_sort(nums):  
n = len(nums)  
for i in range(1, n):  
j = i-1  
print("jota:", j)  
key = nums[i]  
print("key", key)  
print(nums[j],"es mayor que ",key,"?")   
  
while j >= 0 and nums[j] > key:  
print("entro al while")  
nums[j+1] = nums[j]  
j -= 1  
print("jota = ",j)  
nums[j+1] = key  
print(key)  
return nums  
  
insertion\_sort(muns)  
  
ALGORITMO DE ORDENACIÓN MERGE SORT  
  
#muns=[2,5,7,8,74,85,91]  
muns=[12,52,7,18,174,85,91]  
  
def merge\_sort(nums):  
if len(nums) > 1:  
mid = len(nums) // 2  
left\_half = nums[:mid]  
right\_half = nums[mid:]  
print(left\_half)  
print(right\_half)  
  
merge\_sort(left\_half)  
merge\_sort(right\_half)  
i = j = k = 0  
while i < len(left\_half) and j < len(right\_half):  
if left\_half[i] < right\_half[j]:  
print(left\_half[i], "< ",right\_half[j])  
nums[k] = left\_half[i]  
i += 1  
else:  
nums[k] = right\_half[j]  
j += 1  
k += 1  
while i < len(left\_half):  
nums[k] = left\_half[i]  
i += 1  
k += 1  
while j < len(right\_half):  
nums[k] = right\_half[j]  
j += 1  
k += 1  
  
merge\_sort(muns)  
print(muns)