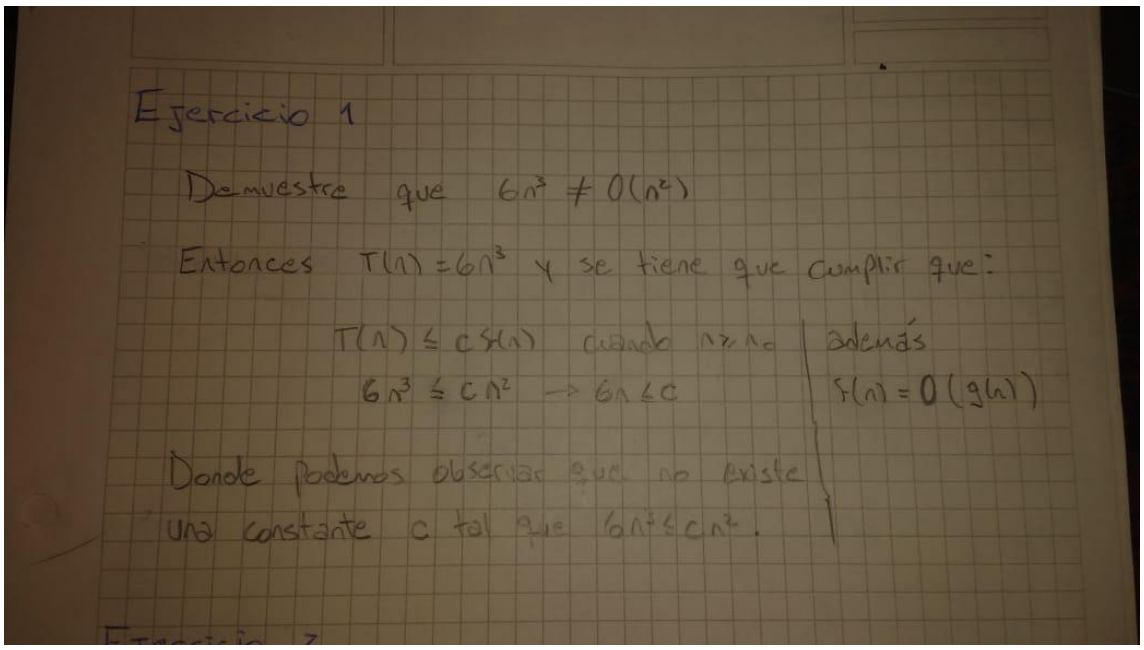
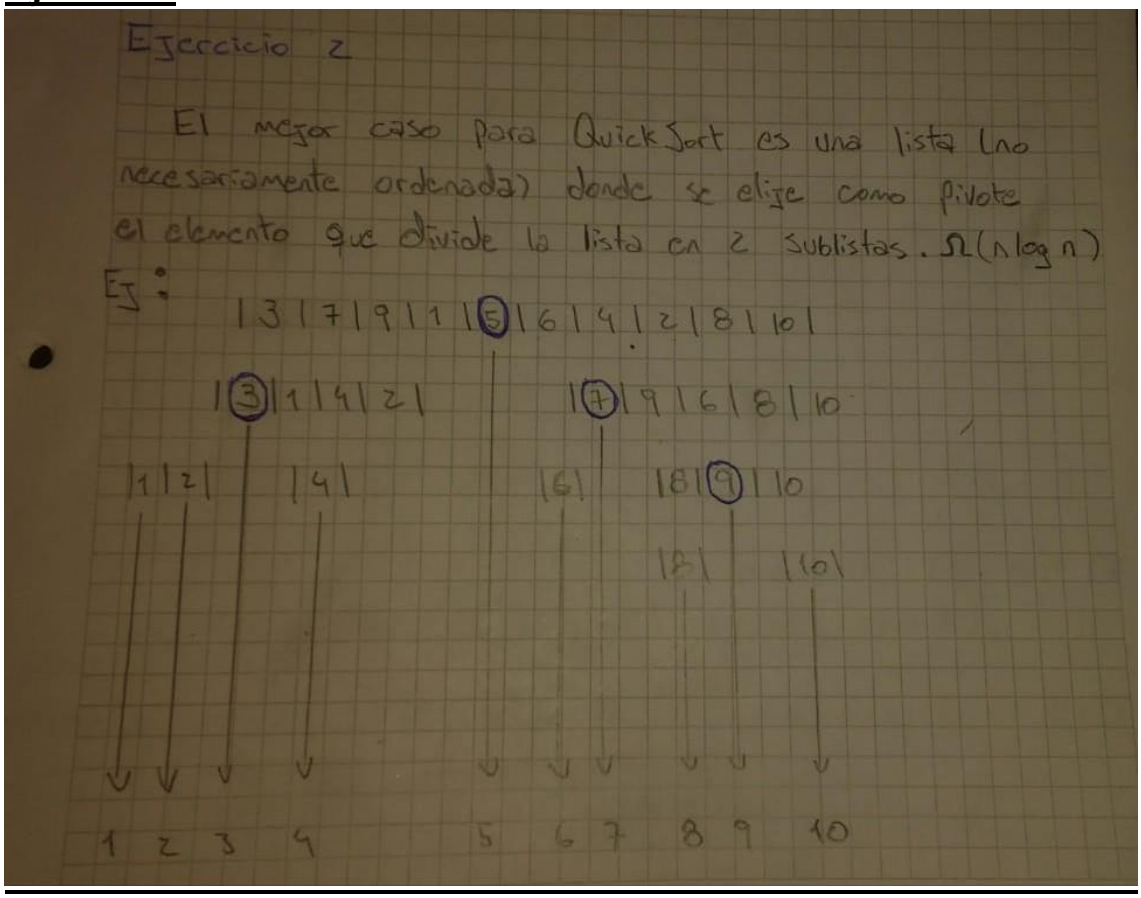
TP Complejidad

Nombre: Victor Ramirez

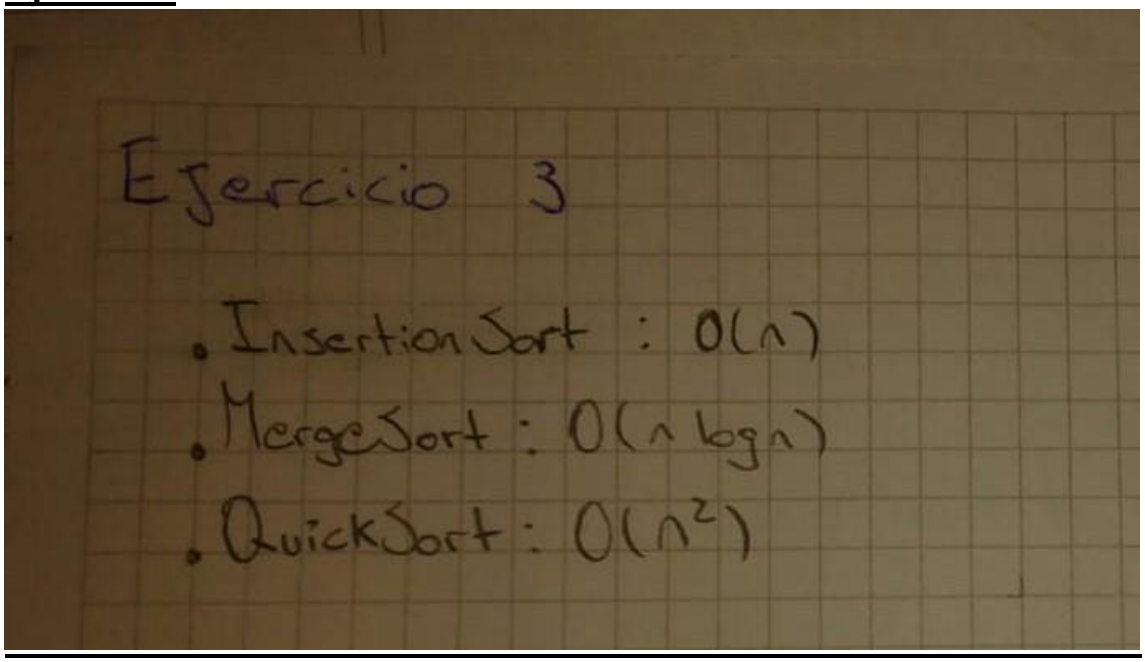
Ejercicio 1



Ejercicio 2



Ejercicio 3



Ejercicio 4

```python
def ordenar_list(list):
    medio = len(list)//2
    aux = list[medio-1]
    izq = list[:medio-1]
    der = list[medio:]
    #print(izq)
    #print(der)
    #print(aux)
    rta = []

    while izq != [] and der != []:

        if izq != []:
            rta.append(izq.pop())
        if der != []:
            rta.append(der.pop())

    rta.insert(medio -1 , aux)
    #print(rta)
    return rta
```

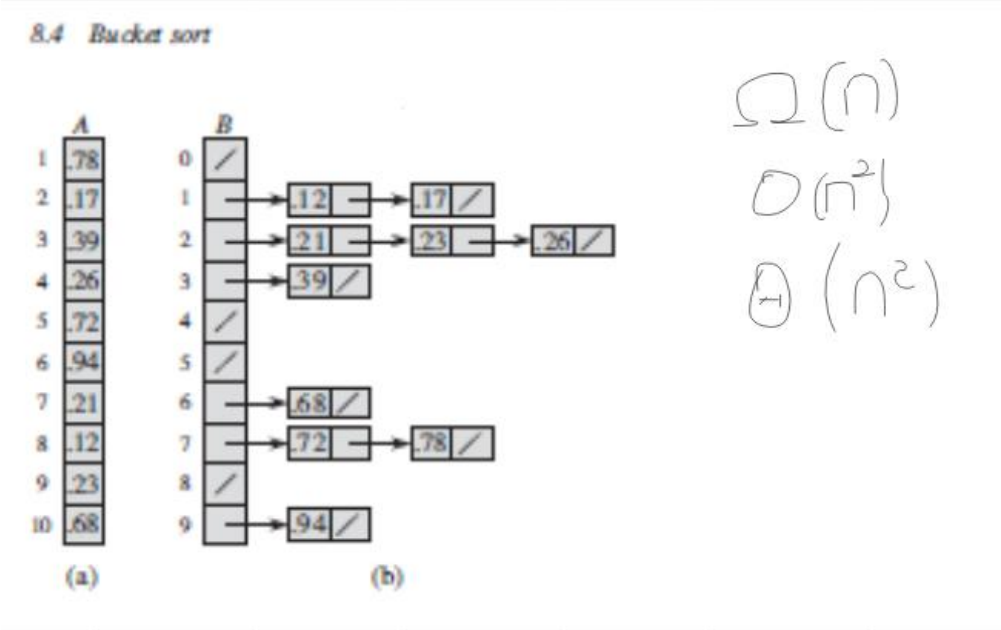
Ejercicio 5

```python
def contiene_suma(A, n):
    A.sort() #O(n logn)
    left = 0
    right = len(A) - 1

    while left < right:
        suma = A[left] + A[right]
        if suma == n:
            return True
        elif suma < n:
            left += 1
        else:
            right -= 1
    return False

#En el peor de sus casos es O(nlogn)
list = [7,3,2,8,5,4,1,6,10,9]
print(contiene_suma(list , 20)) #Caso falso
print(contiene_suma(list,10)) #Caso Verdadero
```

Ejercicio 6

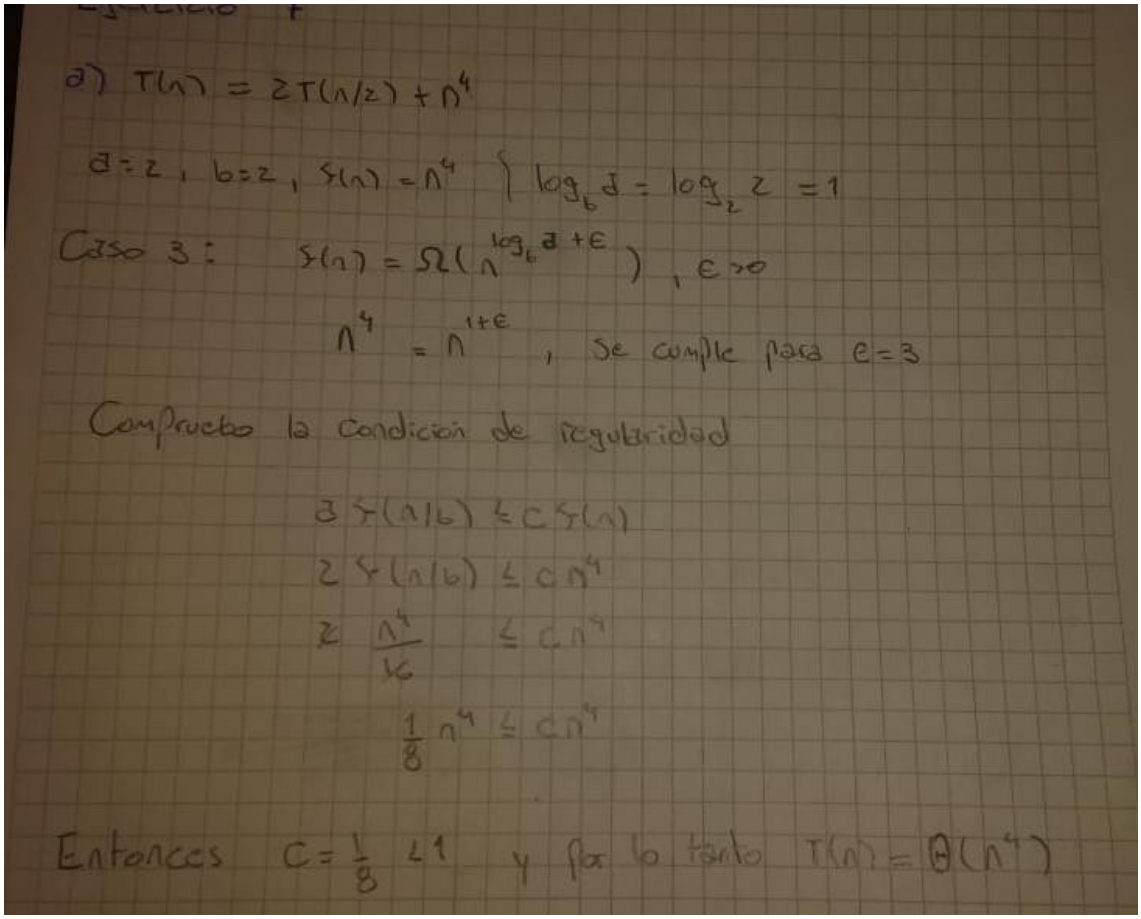
El algoritmo **Bucket sort** es un algoritmo de ordenamiento que divide el conjunto de elementos a ordenar en "buckets" o "cubetas", y luego ordena cada uno de estos buckets por separado utilizando otro algoritmo de ordenamiento, como Insertion sort, Merge sort o Quick sort. Finalmente, se concatenan todos los buckets para obtener la lista ordenada.

Ejemplo:



Si ordenamos las sublistas con Insertion Sort entonces tendremos el mejor caso, caso promedio y peor caso de esa forma. Pero esto puede variar, ya que si ordenamos las sublistas con Merge Sort el peor caso serias O(nlogn).

Ejercicio 7



$\frac{c}{8}$ y por lo tanto $T(n) = O(n$

b) $T(n) = 2T(7n/10) + n$

$a = 2$, $b = 10/7$, $f(n) = n$ | $\log_b a \approx 1{,}94$

Caso 1: $f(n) = O(n^{1{,}94-\epsilon})$, $\epsilon > 0$, se cumple para $\epsilon \approx 0{,}94$

Entonces: $T(n) = \Theta(n^{\log_{10/7} 2})$

c) $T(n) = 16(n/4) + n^2$

$a = 16$, $b = 4$, $f(n) = n^2$ | $\log_4 16 = 2$

Caso 2: $f(n) = \Theta(n^{\log_b a})$

$f(n) = \Theta(n^2)$

Entonces: $T(n) = \Theta(n^2 \log n)$

d) $T(n) = 7T(n/3) + n^2$

$a = 7, \ b = 3, \ c = 2$

Entonces: $\log_b a = \log_3 7 = 1,77 < 2$

Por lo tanto: $T(n) = \Theta(n^2)$

e) $T(n) = 7T(n/2) + n^2$

$a = 7, \ b = 2, \ c = 2$

Ent: $\log_2 7 = 2,8 > 2$

Por lo tanto: $T(n) = \Theta(n^{\log_2 7})$

f) $T(n) = 2T(n/4) + n^{1/2}$

$a = 2, \ b = 4, \ c = \frac{1}{2}$

Ent: $\log_4 2 = \frac{1}{2} = c$

Por lo tanto: $T(n) = \Theta(n^{1/2} \log n)$