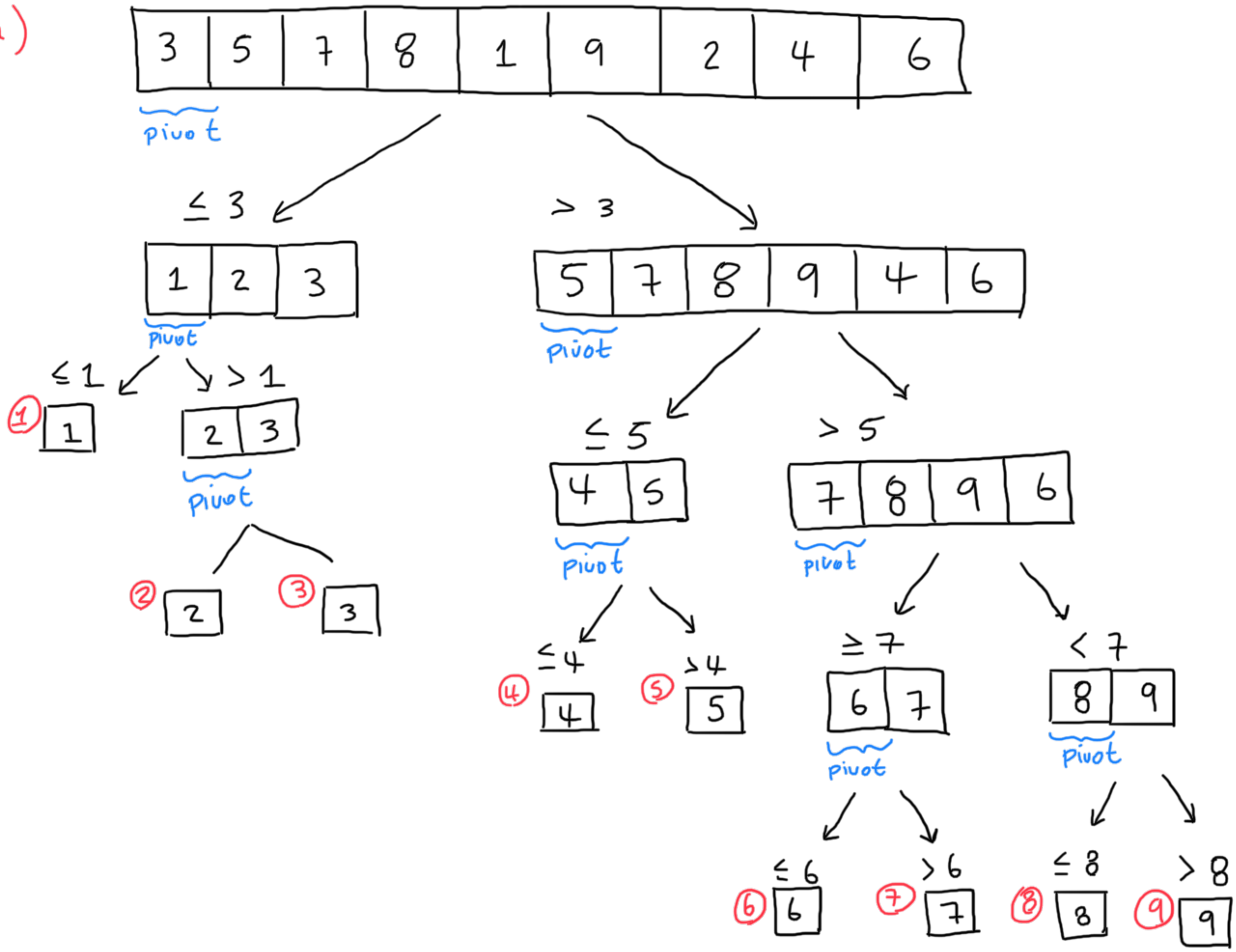
Ejercicio 4 - David Corzo

1 a)



1b)

```
ordered = []
function quicksort(arr[]) {
    if length(arr) ≠ 1 {
        pivot = arr[0]
        arr.remove(pivot)
        low, high = [],[]
        for i in arr {
            if ( i ≤ pivot ) {
                low.append(i)
            } else {
                high.append(i)
            }
        }
```

```
            low.append(pivot)
            quicksort(low)
            quicksort(high)
      } else {
            ordered.append(arr[0])
      }
}
```

1c)

$$\{\ \underbrace{a_1}_{\text{pivot}},\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7,\ \dots,\ a_n\ \}$$

$$\underbrace{\{\ \dots,\dots,\dots,\dots,a_1\ \}}_{\leq a_1} \qquad \underbrace{\{\ \dots,\dots,\dots,\dots,\dots\ \}}_{> a_1}$$

OPERAR El ARRAY PRINCIPAL

```
if length({a1, a2, a3, a4, a5, ..., an}) ≠ 1 { } 1
    pivot = a1 } 1
    arr.remove(pivot) → {a2, a3, a4, ..., an} } 1
    low, high = {},{} } 2
    for i in {a2, a3, a4, ..., an} {
        if (i ≤ pivot) { } 2
            low.append(i) }  ─┐
        } else {              │ 1
            high.append(i) } ─┘
        }
    }                                  n - 1

    low.append(pivot) } 1
```

$= 1 + 1 + 1 + 2 + 2(n-1) + (n-1) + 1 + 1$

$= 7 + 2n - 2 + n - 1$

$= 4 + 3n$

$= \underline{3n + 4}$

quick sort (low)

Esta llamada recursiva manda una lista de tamaño desconocido denotado como K de tal manera que el length del low array es igual $n - K$.

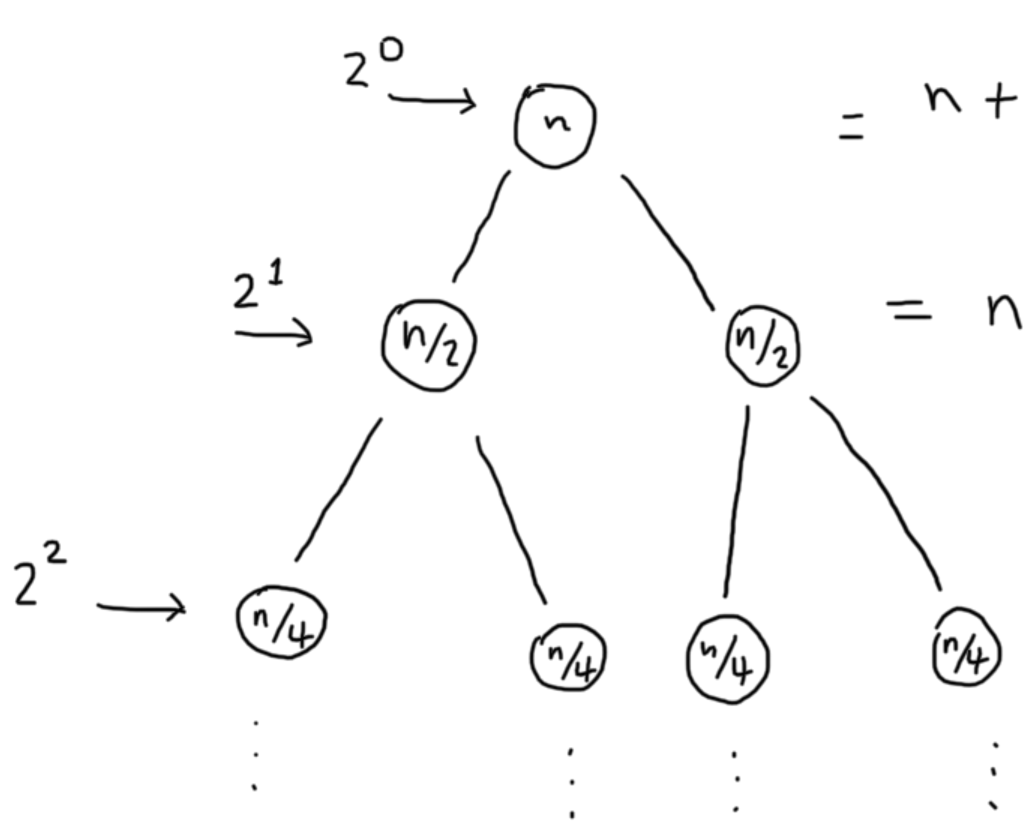
quick sort (high)

Esta llamada recursiva manda una lista de tamaño desconocido pero en proporción $n + K$

R// # En total se compararán n veces más y las comparaciones serían sólo dos veces lo que ya sacamos

$$= 2(3n+4) = \boxed{6n+8}$$

4 d) • Las mínimas comparaciones ocurren cuando low = high, las dos listas tienen el mismo tamaño.



$$= n + \left(\frac{n}{2}\right)2 + \left(\frac{n}{2^2}\right)2^2 + \ldots + \left(\frac{n}{n}\right)n$$

$$= n + n + n + \ldots + n$$

Si $n = 4$:

$$= 4 + \left(\frac{4}{2}\right)2 = 4 + 4$$

$\therefore$ 8 comparaciones mínimas

• Las máximas comparaciones ocurren cuando la lista esta ordenada.

no se operan

si len(arr)==1:

$a_1 \leq a_2 \leq \dots \leq a_n$

$\xrightarrow{4} \boxed{\{a_1, a_2, a_3, a_4\}}$

$\boxed{\{a_1\}}$ $\quad \xrightarrow{3} \boxed{\{a_2, a_3, a_4\}}$

$\boxed{\{a_2\}}$ $\quad \xrightarrow{2} \boxed{\{a_3, a_4\}}$

$\boxed{\{a_3\}}$ $\quad \boxed{\{a_4\}}$

entonces:

$$n + (n-1) + \dots + (n-n) - 1$$

$$= 4 + (4-1) + (4-2) + (4-3) + (4-4)$$

$$= 4 + 3 + 2 + 1 - 1 = 4 + 5 = 9$$

∴ Mínimo de comparaciones son 8 y el máximo son 9.

Evaluando la función $f(n) = 6n + 8$ demostrada en el inciso c:

mínimo:

$$\underbrace{f(4) + 2f\left(\frac{4}{2}\right)}_{\text{2 niveles del Árbol binario}} + 4f\left(\frac{4}{4}\right)$$

$$f(4) + 2f(2) + \cancel{4f(1)}^{\emptyset} = \{6(4)+8\} + 2\{6(2)+8\}$$

$$= 32 + 40$$

$$= 72$$

máximo:

$$\underbrace{f(4) + f(3) + f(2)}_{\text{3 niveles del Árbol binario.}} = \{6(4)+8\} + \{6(3)+8\} + \{6(2)+8\}$$

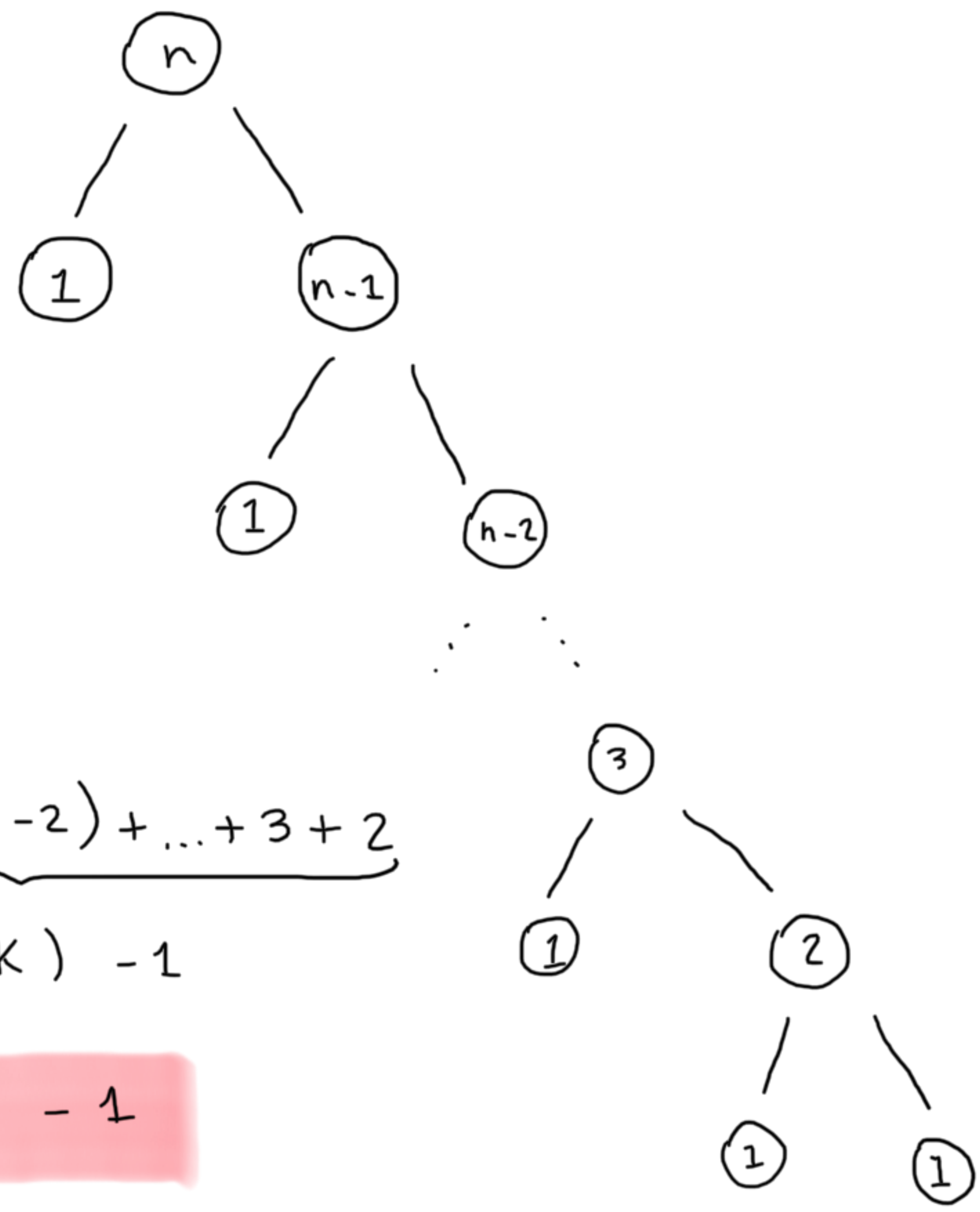
$$= 32 + 26 + 20$$

$$= \underline{\underline{76}}$$

∴ El tiempo para ...

para una lista de 4 esta entre los límites $72 \le t \le 76$.

4e)



$$= \underbrace{n + (n-1) + (n-2) + \ldots + 3 + 2}_{\sum_{k=1}^{n} (k) \; - 1}$$

$$= \frac{(n+1)n}{2} - 1$$

$$= \frac{n^2+n}{2} - 1 \longrightarrow O(n^2)$$

$\therefore$ En el peor de los casos se se tendrá una complejidad de $O(n^2)$.

Comprobación:

$$6n + 8 \le n^2 + 0 \longrightarrow 6n + 8 \le \underbrace{n^2 + n^2}_{2n^2}$$

$$c_1 = 2$$