1. Calculá el orden de complejidad de los siguientes algoritmos:

(a) **proc** *f1*(**in** n : **nat**)
if $n \leq 1$ **then skip**
else
for $i := 1$ **to** 8 **do** *f1*(n **div** 2) **od**
for $i := 1$ **to** n^3 **do** $t := 1$ **od**

(b) **proc** *f2*(**in** n : **nat**)
for $i := 1$ **to** n **do**
for $j := 1$ **to** i **do** $t := 1$ **od**
od
if $n > 0$ **then**
for $i := 1$ **to** 4 **do** *f2*(n **div** 2) **od**

a) El procedimiento toma un natural, por lo que el tamaño de entrada es trivial; y la operación a contar son las asignaciones de la variable *t*.
Defino una función recursiva **r(n)** que calculará la cantidad de asignaciones a la variable *t* que ocurren al ejecutar el procedimiento *f1* con el dato de entrada *n*.

$$\mathrm{r(n)} = \mathrm{ops(f1(n))}$$

$$= \mathrm{ops(if\ ...\ fi)}$$

$$= \begin{cases} \mathrm{ops(skip)} & \text{si } n \geq 1 \\ \mathrm{ops(for\ i := 1\ to\ 8\ do\ ...\ od)} + \mathrm{ops(for\ i := 1\ to\ } n^3 \mathrm{\ do\ ...\ od)} & \text{si } n < 1 \end{cases}$$

$$= \begin{cases} 0 & \text{si } n \geq 1 \\ \sum_{i=1}^{8} \mathrm{ops(f1(n\ div\ 2))} + \sum_{i=1}^{n^3} \mathrm{ops(t := 1)} & \text{si } n < 1 \end{cases}$$

$$= \begin{cases} 0 & \text{si } n \geq 1 \\ \sum_{i=1}^{8} \mathrm{r}(\frac{n}{2}) + \sum_{i=1}^{n^3} 1 & \text{si } n < 1 \end{cases}$$

$$\mathrm{r(n)} = \begin{cases} 0 & \text{si } n \geq 1 \\ 8\ \mathrm{r}(\frac{n}{2}) + n^3 & \text{si } n < 1 \end{cases}$$

donde los componentes *a*, *b*, *k* y *g(n)* son:

a = 8	**b = 2**	**k = 3**	**g(n) = n^3**

∴ como $a = b^k$, luego r(n) es del orden **n^3 log(n)**

b) El procedimiento toma un natural, por lo que el tamaño de entrada es trivial; y la operación a contar son las asignaciones a la variable *t*.
Defino una función recursiva **r(n)** que calculará la cantidad de asignaciones de *t* que ocurren al ejecutar el procedimiento *f2* con el dato de entrada *n*.

$$
\begin{aligned}
\mathrm{r(n)} &= \mathrm{ops(f2(n))} \\
&= \mathrm{ops(for\ i := 1\ to\ } n \mathrm{\ do\ ...\ od)} + \mathrm{ops(if\ ...\ fi)} \\
&= \sum_{i:=1}^{n} \mathrm{ops(for\ j := 1\ to\ i\ do\ ...\ od)} + \begin{cases} \mathrm{ops(for\ i := 1\ to\ 4\ do\ ...\ od)} & \text{si } n > 0 \\ \mathrm{ops(skip)} & \text{si } n \leq 0 \end{cases} \\
&= \sum_{i:=1}^{n} \sum_{j:=1}^{i} \mathrm{ops(t := 1)} + \begin{cases} \sum_{i:=1}^{4} \mathrm{ops(f2(n\ div\ 2))} & \text{si } n > 0 \\ 0 & \text{si } n \leq 0 \end{cases} \\
&= \sum_{i:=1}^{n} \sum_{j:=1}^{i} 1 + \begin{cases} \sum_{i:=1}^{4} \mathrm{r}(\frac{n}{2}) & \text{si } n > 0 \\ 0 & \text{si } n \leq 0 \end{cases} \\
&= \sum_{i:=1}^{n} \mathrm{i} + \begin{cases} 4\ \mathrm{r}(\frac{n}{2}) & \text{si } n > 0 \\ 0 & \text{si } n \leq 0 \end{cases} \\
&= \frac{n(n-1)}{2} + \begin{cases} 4\ \mathrm{r}(\frac{n}{2}) & \text{si } n > 0 \\ 0 & \text{si } n \leq 0 \end{cases}
\end{aligned}
$$

$$
\mathrm{r(n)} = \begin{cases} 4\ \mathrm{r}(\frac{n}{2}) + (\frac{n^2}{2} - \frac{n}{2}) & \text{si } n > 0 \\ \frac{n^2}{2} - \frac{n}{2} & \text{si } n \leq 0 \end{cases}
$$

donde los componentes *a*, *b*, *k* y *g(n)* son:

a = 4	**b = 2**	**k = 2**	**g(n) =** $\frac{n^2}{2} - \frac{n}{2}$

∴ como $a = b^k$, luego r(n) es del orden **n^2 log(n)**