Ejercicio 2

Sea el método siguiente, que en su llamada inicial toma como entrada un número entero *e>0* y un número real *ac*:

```
Double exam(Integer e, Double ac) {
  Double res = 0.;
  if (e <= 2) {
    res = ac/2;
  } else {
    Integer i = 1;
    while(i*i <= e*e) {
      res += i;
      i *=2;
    }
    res = exam(e/2, e + ac) + e/3 - exam(e/4, ac - e);
    while(i > 0) {
      res += i/2 + (e - 3);
      i -= 2;
    }
  }
  return res;
}
```

SE PIDE:

a) Definir el tamaño del problema, n.
b) Definir la ecuación de recurrencia del tiempo de ejecución T(n).
c) Calcular el orden de complejidad del método *exam*.

Tiempo estimado: 40 minutos **Puntuación: ADDA 25%, EDA 33'3%**

SOLUCION

a) El tamaño viene dado por el valor del parámetro e, es decir, n = e.

b) Para la ecuación de recurrencia, resolvemos antes los sumatorios correspondientes a los dos bucles *while*. En toda llamada se cumple que *e>0*, por lo que:

- Para el primer bucle, donde si $i^2 = e^2$ entonces $i = e$, el sumatorio sería:
  $\sum_{pg<(1,2)}^{n} 1 \rightarrow x^0\ log^0 x = 1\ \rightarrow d = 0, p = 0\ \rightarrow \sum_{pg<(1,2)}^{n} 1 \cong_\infty \mathbf{log\, n}$

- Para el segundo bucle tendríamos:
  $\sum_{pa<(n,-2)}^{1} 1 = \sum_{pa<(1,2)}^{n} 1 \rightarrow x^0\ log^0 x = 1\ \rightarrow d = 0, p = 0\ \rightarrow$
  $\sum_{pa<(1,2)}^{n} 1 =_\infty \frac{1}{2(0+1)}\ n^{0+1}\ log^0 n = \boldsymbol{n}$

La ecuación de recurrencia quedaría: $\mathrm{T(n)} = \mathrm{T}\left(\frac{\mathrm{n}}{2}\right) + \mathrm{T}\left(\frac{\mathrm{n}}{4}\right) + \mathrm{n} + \log n$

De la ecuación anterior podemos retirar el último sumando, ya que $\lim_{n\to\infty} \frac{\log n}{n} = 0$, quedando por tanto (#, ##): $\mathrm{T(n)} = \mathrm{T}\left(\frac{\mathrm{n}}{2}\right) + \mathrm{T}\left(\frac{\mathrm{n}}{4}\right) + \theta(n)$

c) Rec. múltiple con distintos tamaños, se aproxima con dos ecuaciones, $T_1(n)$ y $T_2(n)$:

$$\mathrm{T_1(n)} = \mathrm{aT}\left(\frac{\mathrm{n}}{\mathrm{b}}\right) + \mathrm{n}^d \log^p n = \mathbf{2T}\left(\frac{\mathbf{n}}{\mathbf{4}}\right) + \mathbf{n};\ \ \mathrm{a} = 2, \mathrm{b} = 4, \mathrm{d} = 1, \mathrm{p} = 0\ (*)$$
$$\mathrm{T_2(n)} = \mathrm{aT}\left(\frac{\mathrm{n}}{\mathrm{b}}\right) + \mathrm{n}^d \log^p n = \mathbf{2T}\left(\frac{\mathbf{n}}{\mathbf{2}}\right) + \mathbf{n};\ \ \mathrm{a} = 2, \mathrm{b} = 2, \mathrm{d} = 1, \mathrm{p} = 0\ (*)$$

El orden de complejidad queda acotado de la forma:

$$\Omega(\mathrm{n}) = \theta(\mathrm{T_1(n)}) = \mathrm{n^1 log^0 n} = \mathbf{n};\ \ \mathrm{a} = 2 < 4^1 = \mathrm{b}^d$$

$$\mathrm{O(n)} = \theta(\mathrm{T_2(n)}) = \mathrm{n^1 log^{0+1} n} = \mathbf{n\,log}\,\boldsymbol{n};\ \ \mathrm{a} = 2 = 2^1 = \mathrm{b}^d$$

$$\Omega(\mathrm{n}) = \boxed{\boldsymbol{\theta}(\boldsymbol{n}) <_\infty \boldsymbol{\theta}(\mathbf{T(n)}) <_\infty \boldsymbol{\theta}(\mathbf{n\,log}\,\boldsymbol{n})} = \mathrm{O(n)}$$

(*) También: $\mathrm{T_1(n)} = 2\mathrm{T}\left(\frac{\mathrm{n}}{4}\right) + \theta(n);\ \ \mathrm{T_2(n)} = 2\mathrm{T}\left(\frac{\mathrm{n}}{2}\right) + \theta(n);$

(#) La complejidad de una secuencia de bloques donde siempre todos se ejecutan es la mayor de cada una de las complejidades individuales de cada bloque

(##) Considerando que $\forall$ n $\exists\ k_n, 0 < k_n \ll$ n mediante el cual: n $k_n \approx_\infty \mathrm{n} + \log n$, entonces: $\lim_{n\to\infty} \frac{k_n}{n} = 0$