

Algoritmos y Estructuras de Datos - Curso 2011
1er Parcial - Sábado 11 de Junio

Ejercicio 1.

Dado el siguiente algoritmo,

a.- Calcular el T(n)

b.- Calcular el O(n) justificando debidamente.

```
for ( i=1; i<=n; i++) {  
    for ( j=i^2; j>=1; j--) {  
        for ( k=i; k<=j; k++) {  
            ** operacion de orden 1**;  
        }  
    }  
}
```

Pistas:

$$\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6} \quad \sum_{i=1}^n i^3 = \frac{n^2(n+1)^2}{4} \quad \sum_{i=1}^n i^4 = \frac{n(n+1)(6n^3+9n^2+n-1)}{30}$$

Solución:

$$\begin{aligned} \sum_{i=1}^n \left(\sum_{j=1}^{i^2} \left(\sum_{k=i}^j c \right) \right) &= \sum_{i=1}^n \left(\sum_{j=1}^{i^2} (c(j-i+1)) \right) = \sum_{i=1}^n \left(c \sum_{j=1}^{i^2} (j-i+1) \right) = \\ &= \sum_{i=1}^n \left(c \left(\sum_{j=1}^{i^2} j - \sum_{j=1}^{i^2} i + \sum_{j=1}^{i^2} 1 \right) \right) = \sum_{i=1}^n \left(c \left(\frac{i^2(i^2+1)}{2} - i^3 + i^2 \right) \right) = \\ &= c \sum_{i=1}^n \left(\frac{i^2(i^2+1)}{2} - i^3 + i^2 \right) = c \sum_{i=1}^n \left(\frac{i^4}{2} + \frac{i^2}{2} - i^3 + i^2 \right) = \\ &= c \left(\frac{\frac{n(n+1)(6n^3+9n^2+n-1)}{30}}{2} + \frac{\frac{n(n+1)(2n+1)}{6}}{2} - \frac{n^2(n+1)^2}{4} + \frac{n(n+1)(2n+1)}{6} \right) = \\ &= c \left(\frac{n(n+1)(6n^3+9n^2+n-1)}{60} + \frac{n(n+1)(2n+1)}{12} - \frac{n^2(n+1)^2}{4} + \frac{n(n+1)(2n+1)}{6} \right) = \\ &= c \left(\frac{(n^2+n)(6n^3+9n^2+n-1)}{60} + \frac{(n^2+n)(2n+1)}{12} - \frac{n^2(n^2+2n+1)}{4} + \frac{(n^2+n)(2n+1)}{6} \right) = \\ &= c \left(\frac{6n^5+9n^4+n^3-n^2+6n^4+9n^3+n^2-n}{60} + \frac{2n^3+n^2+2n^2+n}{12} - \frac{n^4+2n^3+n^2}{4} + \frac{2n^3+n^2+2n^2+n}{6} \right) = \end{aligned}$$

$$\begin{aligned}
&= c \left(\frac{6}{60}n^5 + \frac{9}{60}n^4 + \frac{1}{60}n^3 - \frac{1}{60}n^2 + \frac{6}{60}n^4 + \frac{9}{60}n^3 + \frac{1}{60}n^2 - \frac{1}{60}n + \frac{2}{12}n^3 + \frac{1}{12}n^2 + \frac{2}{12}n^2 + \frac{1}{12}n - \frac{1}{4}n^4 - \frac{2}{4}n^3 \right) = \\
&= c \left(\frac{6}{60}n^5 + \frac{9}{60}n^4 + \frac{6}{60}n^4 - \frac{1}{4}n^4 + \frac{1}{60}n^3 + \frac{9}{60}n^3 + \frac{2}{12}n^3 - \frac{2}{4}n^3 + \frac{2}{6}n^3 - \frac{1}{60}n^2 + \frac{1}{60}n^2 + \frac{1}{12}n^2 + \frac{2}{12}n^2 - \frac{1}{4}n^2 \right) = \\
&= c \left(\frac{6}{60}n^5 + \frac{9}{60}n^4 + \frac{6}{60}n^4 - \frac{15}{60}n^4 + \frac{1}{60}n^3 + \frac{9}{60}n^3 + \frac{10}{60}n^3 - \frac{30}{60}n^3 + \frac{20}{60}n^3 - \frac{1}{60}n^2 + \frac{1}{60}n^2 + \frac{5}{60}n^2 + \frac{10}{60}n^2 - \frac{15}{60}n^2 \right) = \\
&= c \left(\frac{6}{60}n^5 + \frac{10}{60}n^3 + \frac{20}{60}n^2 + \frac{14}{60}n \right) =
\end{aligned}$$

$$\frac{6}{60}cn^5 + \frac{10}{60}cn^3 + \frac{20}{60}cn^2 + \frac{14}{60}cn - es - O(n^5) \Rightarrow \frac{6}{60}cn^5 + \frac{10}{60}cn^3 + \frac{20}{60}cn^2 - \frac{14}{60}cn \leq kn^5 \forall n \geq n_0 \Rightarrow$$

$$\frac{6}{60}cn^5 \leq k_1n^5, k_1 = \frac{6}{60}c, \forall n_0,$$

$$\frac{10}{60}cn^3 \leq k_2n^5, k_2 = \frac{10}{60}c, \forall n_0$$

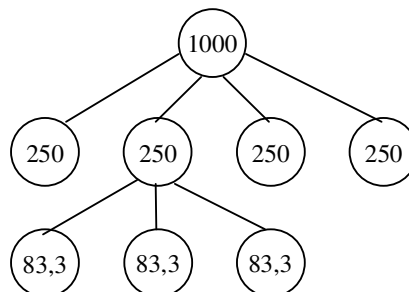
$$\frac{20}{60}cn^2 \leq k_3n^5, k_3 = \frac{20}{60}c, \forall n_0$$

$$\frac{14}{60}cn \leq k_4n, k_4 = \frac{14}{60}c, \forall n_0$$

$$\frac{6}{60}cn^5 + \frac{10}{60}cn^3 + \frac{20}{60}cn^2 + \frac{14}{60}cn \leq kn^5, k = \left(\frac{6}{60}c + \frac{10}{60}c + \frac{20}{60}c + \frac{14}{60}c \right), \forall n_0$$

Ejercicio 2.-

Sea una red de agua potable, la cual comienza en un caño maestro y el mismo se va dividiendo sucesivamente hasta llegar a cada una de las casas. Por el caño maestro ingresan 1000 litros y en la medida que el caño se divide, el caudal se divide en partes iguales en cada una de las divisiones. Es decir, si el caño maestro se divide en 4 partes, cada división tiene un caudal de 250 litros. Luego, si una de esas divisiones se vuelve a dividir en 3 partes, cada una tendrá un caudal de 83,3. La situación descripta se puede modelar de la siguiente forma a través del siguiente árbol general:



Usted debe implementar un método en la clase árbol general, que considerando que ingresan n litros por el caño maestro, calcule cual es el mínimo caudal que recibe una hoja.

Solución:

```

double minimoCaudal (double caudalEntrante) {
    double answer = caudalEntrante;
    ListaGenerica<ArbolGeneral<T>> hijos = this.getHijos();
    if (!hijos.tamano()==0) {
        hijos.comenzar()
    }
}

```

```

        while (!hijos.fin()) {
            answer = min(answer, hijos.elemento().minimoCaudal( caudalEntrante /
hijos.tamano()));
            hijos.proximo();
        }
    }
    return (answer);
}

```

Ejercicio 3.

Sea una red de empresas, en donde una empresa brinda servicios a muchas otras empresas sin ningún tipo de restricciones. Esta relación no es simétrica, es decir, la empresa A puede brindar servicios a la empresa B, sin embargo, B no necesariamente debe brindar servicios a la empresa A.

Usted es contratado por una nueva empresa que quiere ingresar a la Red. Por lo cual, le interesará determinar las 5 empresas de la red que llegan a la mayor cantidad de empresas. Una empresa A llega a otra empresa B, si le brinda servicios directa o indirectamente a través de otras empresas.

Solución:

Debe hacer un DFS para cada una de las empresas de la red y contar los nodos a los que llega. Para ello, no debe olvidar marcar los nodos en la medida que visita, para evitar quedar en loop y contar varias veces el mismo nodo. Como el DFS se ejecuta una vez por cada nodo, es importante que antes de cada ejecución los nodos se vuelvan a marcar como no visitados.