TEMA 4: DISEÑO DE ALGORITMOS RECURSIVOS

1. Principio de inducción
2. Diseño recursivo
3. Coste temporal de un algoritmo recursivo
4. Coste espacial de un algoritmo recursivo
5. Inmersión de programas
6. Inmersión de especificaciones

1. Principio de inducción

- **El principio de inducción matemática es fundamental en la informática.**
- **En esta asignatura lo utilizaremos para verificar funciones recursivas y la estructura iterativa. También lo utilizaremos para demostrar propiedades sobre los TAD.**
- **Vamos a ver 3 tipos de inducción diferentes:**
  - Débil
  - Fuerte
  - Inducción general o noetheriana

1. Principio de inducción

- **Principio de inducción débil:**

  Def: Sea P: $\mathcal{N} \rightarrow \mathcal{B}$ una propiedad definida sobre $\mathcal{N}$.

  $[P(0) \wedge (P(k-1) \Rightarrow P(k)\ \forall\ k > 0, k \in N)] \Rightarrow P(n)\ \forall\ n \in \mathcal{N}$

  - **Base de inducción (BI):** Demostración de P(0).
  - **Hipótesis de inducción (HI):** P(k – 1)
  - **Paso de inducción (PI):** Demostración de $P(k-1) \Rightarrow P(k)\ \forall\ k > 0, k \in \mathcal{N}$

  Ejercicio 1: Demostrar $n^2 = \sum \alpha: 1 \leq \alpha \leq n: (2\alpha - 1)$

  Ejercicio 2: Demostrar que $10^n - 1$ es divisible por 9 $\forall\ n \in N, n>0$.

1. Principio de inducción

- **Principio de inducción fuerte:**

  Def: Sea P: $\mathcal{N} \rightarrow \mathcal{B}$ una propiedad definida sobre $\mathcal{N}$.

  $[P(0) \wedge (\,(P(l)\ \forall\ l < k) \Rightarrow P(k)\ \forall\ k \in N)\,] \Rightarrow P(n)\ \forall\ n \in \mathcal{N}$

  Ejercicio: Demostrar que $F(n) = 1 + \lg n$ siendo F(n):

  $$F(n) = \begin{cases} 1 & n = 1 \\ F(n/2) + 1 & n > 1 \end{cases}$$

1. Principio de inducción

- **Principio de inducción general o noetheriano:**
  - Para dominios que no sean naturales (pilas, árboles, ...)

Def (Preorden): Una relación binaria en D, $\preccurlyeq \subseteq D \times D$ se dice que es un preorden en D si es reflexiva y transitiva.

reflexiva: $\forall\ x \in D, x \preccurlyeq x$

transitiva: $\forall\ x, y, z \in D, x \preccurlyeq y \wedge y \preccurlyeq z \Rightarrow x \preccurlyeq z$

Cuando un preorden cumple además la propiedad antisimétrica, se dice que es un orden parcial.

antisimétrica: $\forall\ x, y \in D, x \preccurlyeq y \wedge y \preccurlyeq x \Rightarrow x = y$

1. Principio de inducción

Def (preorden estricto): Dado un preorden $\preccurlyeq$ en D, definimos la relación $\prec$, que llamaremos preorden estricto, como:

$x \prec y \equiv x \preccurlyeq y \wedge \neg(y \preccurlyeq x)$

El preorden estricto cumple las propiedades transitiva y antirreflexiva.

antirreflexiva: $\forall\ x \in D, \neg(x \prec x)$

Def (elemento minimal): Dado un preorden $\preccurlyeq$ en D, se dice que un elemento $m \in D$ es minimal si no tiene predecesores estrictos:

$\neg(\exists\ x \in D, x \prec m)$

1. Principio de inducción

Def (preorden bien fundado o pbf): Un preorden $\preccurlyeq$ en D, se dice que es bien fundado si no existen en D sucesiones infinitas estrictamente decrecientes, es decir:

$\{x_i\}$ tales que $\forall\ i \in N,\ x_{i+1} \prec x_i$

Un ejemplo de preorden bien fundado es la relación $\leq$ de los naturales o también el orden lexicográfico.

1. Principio de inducción

Teorema: Si $\leq_2$ es un pbf en D2, D1 un conjunto, f: D1 $\rightarrow$ D2 una aplicación y a, b $\in$ D1 y definimos a $\leq_1$ b $\equiv$ f(a) $\leq_2$ f(b). Entonces $\leq_1$ es un pbf en D1.

Ejemplo pbf sobre pilas:

$\forall$ p1, p2 $\in$ pila, p1 $\preccurlyeq$ p2 $\equiv$ altura(p1) $\leq$ altura(p2)

Hay ocasiones en donde no se puede encontrar dicha aplicación, aunque ello no significa que no sea un pbf (Ejemplo: orden lexicográfico).

1. Principio de inducción

Teorema (Principio de inducción noetheriano): Sea $\preccurlyeq$ un pbf en D, P(x) una propiedad sobre los elementos x de D y M el conjunto de los elementos minimales en D (M = {m: -($\exists$ x $\in$ D, x $\prec$ m)})

$[P(m)\ \forall\ m \in M \wedge (\ (P(l)\ \forall\ l \prec k) \Rightarrow P(k)\ \forall\ k \in D)\ ] \Rightarrow P(n)\ \forall\ n \in D$

- **Teoremas inductivos sobre TAD:** El principio de inducción noetheriano lo utilizaremos para demostrar propiedades de los TAD. Para garantizar que tenemos un pbf utilizaremos normalmente la función de tamaño del tipo.

1. Principio de inducción

Ejercicio: Demostrar que para toda cola distinta de c_nula se cumple

suma(c) $\equiv$ primero(c) + suma(avance(c))

Ecuaciones de la suma de una cola:

1) suma(c_nula) $\equiv$ 0

2) suma(pide_turno(n, c)) $\equiv$ n + suma(c)