

UNITAT 7

RECURSIVITAT

EXEMPLES

PROGRAMACIÓ
CFGs DAW

Autors:

Guillermo Garrido Portes – g.garridoportes@edu.gva.es

2023/2024

Llicència



CC BY-NC-SA 3.0 ES Reconeixement – No Comercial – Compartir Igual (by-nc-sa) No es permet un ús comercial de l'obra original ni de les possibles obres derivades, la distribució de les quals s'ha de fer amb una llicència igual a la que regula l'obra original. NOTA: Aquesta és una obra derivada de l'obra original realitzada per Carlos Cacho i Raquel Torres.

Càlcul de factorial

El factorial d'un nombre enter n , representat com a $n!$, es defineix com el producte de tots els nombres enters des d'1 fins a n . Per exemple: $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$

```
// Versió recursiva
public static int factorial(int n) {
    if (n == 1) {
        return 1;
    } else {
        return n * factorial(n-1);
    }
}
```

```
-----
// Versió iterativa
public static int factorial(int n) {
    int factorial = 1;
    for(int i = 1; i <= n; i++) {
        factorial *= i;
    }
    return factorial;
}
```

Solució recursiva:

- Caso base: Si n és igual a 1, es retorna 1 (el factorial d'1 és 1).
- Caso general: Es fa una anomenada recursiva calculant el factorial de $n-1$ i es multiplica eixe resultat per n .

Per exemple, per a calcular $5!$ es té:

1. factorial(4) crida a factorial(3)
2. factorial(3) crida a factorial(2)
3. factorial(2) flama a factorial(1) ---> retorna 1
4. factorial(2) retorna $1 * 2 = 2$
5. factorial(3) retorna $2 * 3 = 6$
6. factorial(4) retorna $6 * 4 = 24$
7. factorial(5) retorna $24 * 5 = 120$

Solució iterativa:

- S'usa un bucle `*for` per a iterar des d'1 fins a n , calculant el producte en cada pas.

Serie de Fibonacci

La sèrie de Fibonacci comença amb els números 0 i 1, i cada número següent és la suma dels dos números anteriors. Per exemple: 0, 1, 1, 2, 3, 5, 8, 13, 21, ...

Formalment, es defineix com:

$$F(0) = 0$$

$$F(1) = 1$$

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

```
// Versió recursiva
public static int fibonacci(int n) {
    if (n <= 1)
        return n;
    return fibonacci(n-1) + fibonacci(n-2);
}
```

```
-----
// Versió iterativa
public static int fibonacci(int n) {
    if (n <= 1)
        return n;
    int a = 0, b = 1;
    for (int i = 2; i <= n; i++) {
        int tmp = b;
        b = a + b;
        a = tmp;
    }
    return b;
}
```

Solució recursiva:

- Casos base: $F(0)=0$ i $F(1)=1$
- Cas general: Suma els dos números anteriors cridant *recursivament a la funció.

Solució iterativa:

- Inicialitza dues variables amb els valors 0 i 1.
- Usa un bucle per a anar calculant i desplaçant els valors.

Suma de dígit

Donat un nombre enter n , hem de calcular la suma dels seus dígit individuals.

Per exemple: si $n = 5324$, la suma dels seus dígit és $5 + 3 + 2 + 4 = 14$.

```
// Versió recursiva
public static int sumaDigits(int x) {
    if (x < 10)
        return x;
    return x % 10 + sumaDigits(x / 10);
}
```

```
-----
// Versió iterativa
public static int sumaDigits(int x) {
    int suma = 0;
    while (x > 0) {
        suma += x % 10;
        x /= 10;
    }
    return suma;
}
```

Solució recursiva:

- Case base: Si $n < 10$, es retorna n (un sol dígit).
- Cas general:
 1. S'extrau el dígit menys significatiu amb el mòdul (%)
 2. Es divideix entre 10 per a eliminar eixe dígit
 3. Se suma el dígit extret amb una anomenada recursiva a la resta del número

Solució iterativa:

- Bucle while per a anar extraient i sumant cada dígit en cada iteració.
- Es divideix entre 10 en cada pas per a eliminar el dígit ja processat.

Torres de Hanói

El problema consisteix a moure n discos d'un pilar origen a un pilar destí, passant per un pilar auxiliar, subjecte a estes restriccions:

- Només es pot moure un disc alhora.
- Mai es pot col·locar un disc major sobre un menor.

```
// Versió recursiva
public static void hanoi(int n, char origen, char
aux, char destí) {
    if (n == 1)
        mover(origen, destí);
    else {
        hanoi(n-1, origen, destí, aux);
        mover(origen, destí);
        hanoi(n-1, aux, origen, destí);
    }
}

-----
// No te versió iterativa sencilla
```

Solució recursiva:

- Cas base: Si hi ha només 1 disc, es mou directament.
- Cas general:
 1. Moure $n-1$ discos de l'origen a l'auxiliar (recursivament)
 2. Moure el disc més gran al destí
 3. Moure els $n-1$ discos sobre ell, de l'auxiliar al destí (recursivament)