

## 5. Funciones recursivas

**Objetivos:** Usar recursión, observar ventajas y inconvenientes, análisis del flujo del programa, uso de estructuras de datos simples para almacenar resultados intermedios y aliviar el cálculo.

1. Escribe una función recursiva que calcule el factorial de un número positivo entero.
  - Realiza un b́ucle principal para visualizar resultados hasta que se quiere terminar (por ejemplo, introduciendo un número negativo).
  - Experimenta hasta que valores se puede calcular en un tiempo que consideras razonable.

2. Escribe una función recursiva que calcule un coeficiente binomial  $\binom{n}{k}$  según la fórmula

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

- Pide los dos valores  $n$  y  $k$  al usuario.
  - Realiza un b́ucle principal para visualizar resultados hasta que se quiere terminar (por ejemplo, introduciendo un número negativo para  $n$ ).
  - Haz unos experimentos hasta que valores python es capaz de calcular los coeficientes binomiales con este método en un tiempo que consideras razonable (¿cuál es la  $k$  respecto a la  $n$  que produce el coeficiente más grande?).
3. Escribe una función que calcule el  $n$ -ésimo valor de la serie de Fibonacci (recuerda, la serie comienza: 1,1,2,3,5,8,13,21,34,...). Quitando los dos casos iniciales  $f_0 = 1$  y  $f_1 = 1$ , la fórmula recursiva es  $f_n = f_{n-1} + f_{n-2}$ .
    - Pide el valor  $n$  al usuario.
    - Realiza un b́ucle principal para visualizar resultados hasta que se quiere terminar (por ejemplo, introduciendo un número negativo para  $n$ ).
    - Haz unos experimentos hasta que valores python es capaz de calcular la serie con este método en un tiempo que consideras razonable.
  4. Analiza las llamadas que realizan tus programas a las diferentes funciones, es decir, busca una respuesta a la pregunta: ¿Cuántas veces se llama a la función recursiva con un parámetro determinado?

Mejora los programas anteriores considerablemente (respecto al tiempo de cálculo) usando una estructura de datos simple para almacenar valores ya calculados que solamente hay que buscar en la estructura en vez de recalculados.

5. Usa el módulo `timeit` y su documentación disponible en la red para medir más exacto el tiempo de cálculo de tus funciones.