Recursividad

¿Qué es la Recursivad?

Autor 1: Jhon Eyder González Castro

*Risaralda, Universidad Tecnológica de Pereira, Pereira, Colombia*

Correo-e: jhon.gonzalez1@utp.edu.co

***Resumen*— La Recursividad significa "definir un problema en términos de sí mismo". Esta puede ser una herramienta muy poderosa para escribir algoritmos. La recurrencia viene directamente de las matemáticas, donde hay muchos ejemplos de expresiones escritas en términos de sí mismas. Por ejemplo, la secuencia Fibonacci se define como: F(i) = F(i-1) + F(i-2)**

***Palabras clave—* Problema, Serie o Sucesión de Fibonacci, Algoritmos, Función, matemáticas.**

***Abstract*— Recursion means "defining a problem in terms of itself". This can be a very powerful tool in writing algorithms. Recursion comes directly from Mathematics, where there are many examples of expressions written in terms of themselves. For example, the Fibonacci sequence is defined as: F(i) = F(i-1) + F(i-2)**

***Key Word* —- Problem, Fibonacci Succession, Algorithm, Function, math.**

1. INTRODUCCIÓN

La recursividad es el proceso de definir un problema (o la solución a un problema) en términos de (una versión más simple de) sí mismo.

Por ejemplo, podemos definir la operación "encuentra tu camino a casa" como:

Si está en su casa, deje de moverse.

Da un paso hacia la casa.

"Encuentra tu camino a casa".

Aquí la solución para encontrar el camino a casa son dos pasos (tres pasos). Primero, no nos vamos a casa si ya estamos en casa. En segundo lugar, hacemos una acción muy simple que hace que nuestra situación sea más fácil de resolver. Finalmente, rehacemos todo el algoritmo.

1. CONTENIDO

Qué es la recursividad en la programación? No es un algoritmo, ni una estructura de datos. Es una forma de resolver un problema. Es una forma de pensar. Es un proceso en el que una función se llama a sí misma directa o indirectamente recursión, y la función correspondiente se llama función recursiva.

La recurrencia es un concepto bastante simple. Algo que se repite, se refiere a sí mismo. El reflejo en un espejo de un espejo es recursivo: el espejo reflejado refleja su propia imagen y lo hace indefinidamente.

También es probable que hayas visto el efecto Droste al usar videocámaras conectadas a un televisor o a una computadora: cuando la videocámara está mirando a la pantalla, puedes ver que se generan infinitas series de pantallas, ya que la videocámara está grabando la misma imagen que está enviando a la pantalla.

El juego Portal es un gran ejemplo de recursividad, cuando dos portales podían abrirse uno al lado del otro en un espacio estrecho y mirando en cualquiera de ellos se producía una serie infinita de la misma imagen.

Ejemplo 1: muñeca rusa

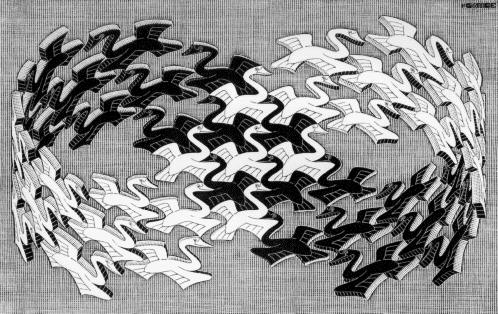


¿Qué tienen que ver las muñecas rusas con los algoritmos?

Así como una muñeca rusa tiene dentro una muñeca rusa más pequeña, que tiene una muñeca rusa aún más pequeña, hasta llegar a una muñeca rusa demasiado pequeña para contener otra, veremos cómo diseñar un algoritmo para resolver un problema resolviendo una instancia más pequeña del mismo problema, a menos que el problema sea tan pequeño que podamos resolverlo directamente.

A esta técnica la llamamos recursión.

Ejemplo 2: Los muchos dibujos de MC Escher.



Escher combinó la recursión y la repetición de patrones de una manera única.

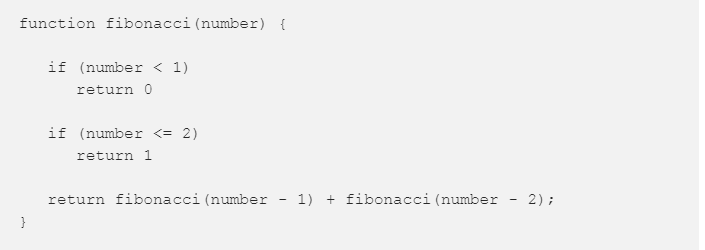
Algunas de las obras que presentan esta combinación exhiben algunas ideas matemáticas y físicas complejas, pero para el espectador casual las obras son sublimes.

La imagen de los cisnes de arriba muestra este tipo de combinación. Nótese que los cisnes están embaldosados de forma muy precisa, con la misma distancia de los cisnes adyacentes y de los cisnes de la siguiente fila.

Nótese también que están en un bucle cerrado, una construcción posible gracias a la recursión.

Ejemplo 3:

Veamos el clásico algoritmo de secuencia de Fibonacci con el que todos estamos familiarizados.



¿Qué pasa dentro?

C:\Users\utp.INFERNOCRIE\Desktop\Image.png

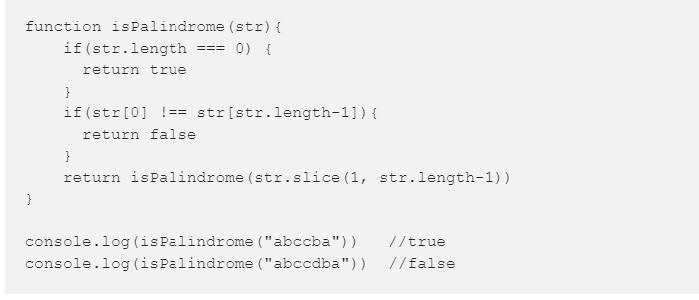
Un ejemplo del mundo real del algoritmo anterior en funcionamiento es la filotaxis, el bello arreglo de las hojas, o los flósculos en algunas plantas.

Las flores de la cabeza de un girasol forman dos espirales opuestas: 55 de ellas en sentido horario y 34 en sentido antihorario.

Estos números son números Fibonacci consecutivos.

Ejemplo 4: ¡Palíndromo o no!

La comprobación de si una cadena es un palíndromo también se puede hacer con recursión.



Ejemplo 5: ¡Un árbol fractal!

Para este ejemplo necesitaremos importar la librería p5.js. En sus palabras--

p5.js es una librería Javascript que comienza con el objetivo original de Processing, hacer accesible la codificación para artistas, diseñadores, educadores y principiantes, y reinterpreta esto para la web actual.

Debo aclarar que esto no es un tutorial sobre la biblioteca p5. Es una demostración de un árbol dibujado recursivamente con p5.js.

Daniel Shiffman de The Processing Foundation y desarrolla tutoriales, ejemplos y bibliotecas para Processing y p5.js. También es muy activo en youtube.

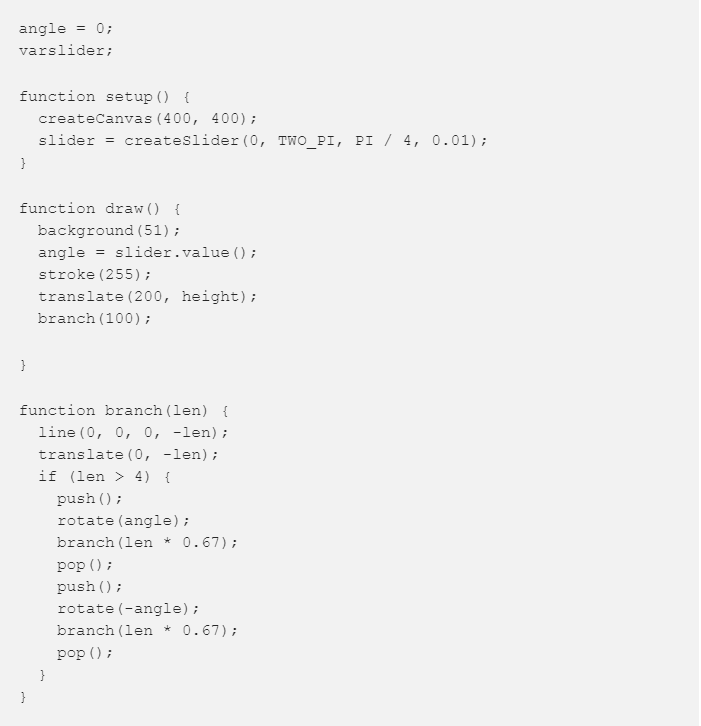
Entonces, ¿cómo dibujamos un árbol fractal. En realidad es bastante fácil. El algoritmo es el siguiente.

Dibujamos el tronco del árbol.

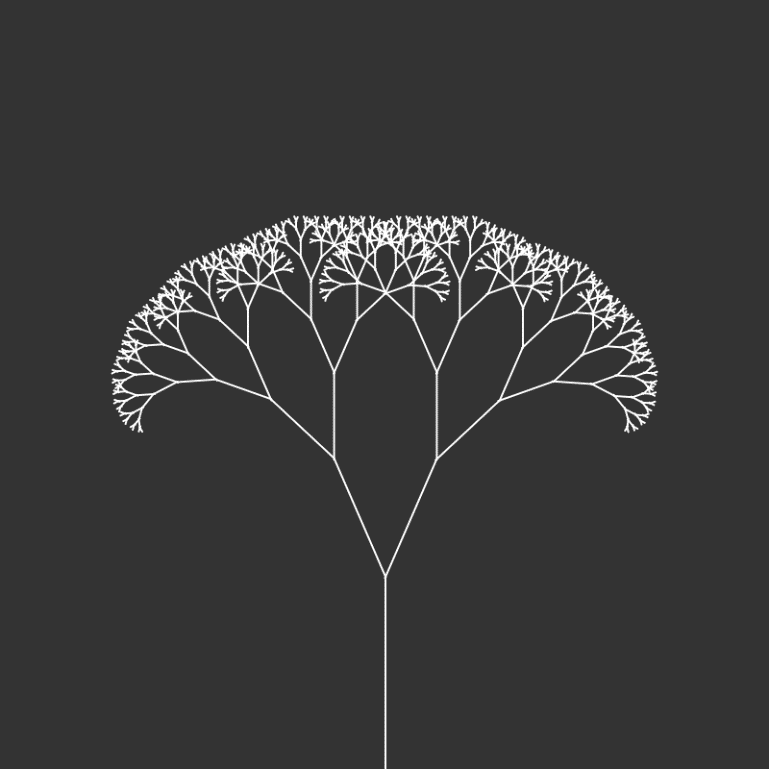
En el punto final de ese tronco, rotamos hasta cierto punto (en este caso 45 grados) y dibujamos una rama.

Luego tenemos que volver a la ubicación de origen de la rama, o al punto final de la rama anterior.

Giramos y dibujamos la misma rama de nuevo.



Cuando todo sale bien, se ve así:



Entonces, ¿qué es la recursión? Una recursión es cuando algo se define por sí mismo.

Todos nos encontramos antes con Factorial en matemáticas. Es una definición recursiva. Factorial de 4 es 4 x 3 x 2 x 1. O, 4! puede ser definido por 4 x 3!

Esto significa que para cualquier número n, su factorial es n x (n-1!).

Hay dos tipos de recursividad.

Una función que se invoca a sí misma.

La función también incluirá un caso base, para que no se produzca un bucle infinito. En el caso de factorial de 4, paramos cuando llega a 1.

Entonces, ¿cuándo es útil en la programación? No siempre, pero algunos algoritmos extremadamente eficientes están diseñados con recursión. Merge sort utiliza la recursión para su implementación.

Muchas otras pueden implementarse con recursividad. Algunos se verán afectados por la actuación y otros podrán mejorar.

Como he mencionado antes, es una forma de resolver problemas, no una bala mágica.

1. CONCLUSIONES

La recursividad está presente en muchos casos de la vida cotidiana y también en muchos campos, por ejemplo está presente en la matemática, en los dibujos, en las ciencias de la computación etc.

Existe un videojuego que se llama Portal, en el que el personaje del videojuego utiliza una pistola que abre dos tipos de portales, un portal naranja y un portal azul, y los lanza en paredes, y al pasar sobre estos se crea un bucle infinito, eso es recursividad.

REFERENCIAS

Referencias Páginas Web:

<https://www.cs.utah.edu/~germain/PPS/Topics/recursion.html>

<https://medium.com/front-end-weekly/a-quick-guide-to-recursion-by-example-c0e7949b8ab6>