TP U5 - Recursión

Alumno: Nuñez Souto, Aaron Agustín

1. El concepto de Recursión hace referencia al acto de una función llamándose a sí misma. La recursión es utilizada para resolver problemas que contienen subproblemas más pequeños. Una función recursiva puede recibir 2 entradas: un caso base (finaliza la recursión) o un caso recursivo (continúa la recursión).
2. Por función recursiva entiendo aquella función que, dentro de sí misma, se invoca nuevamente, o invoca otra función que vuelve a llamar a la primera. Una recurrencia es una función matemática bien definida donde la función que se está definiendo se aplica dentro de su propia definición.
3. Las pilas del sistema operativo funcionan, a la hora de hablar de recursión, añadiendo las invocaciones a cierta función en el orden en que fueron llamadas, por ende, una vez que se invoca por última vez, se comienzan a resolver desde la última invocada hasta la primera. Un ejemplo sería una función factorial la cual, al pasarle un número por parámetro, va a calcular el factorial del mismo invocándose a sí misma hasta llegar al valor mínimo. De esta forma se calcularía de menor a mayor, dado que la última invocación de la función recursiva sería la que contiene el valor mínimo, luego terminaría con la primera función invocada, la cual contiene el valor original pasado por parámetro, retornando finalmente el resultado final.
4. Hay 5 tipos de recursión. La primera es la **recursión directa**, la cual ocurre cuando una función se llama a sí misma. La segunda es la **recursión indirecta**, la cual sucede cuando una función llama a otra que, a su vez, llama de nuevo a la primera. La tercera es la **tail recursion**, la cual se denomina así cuando no hay operaciones pendientes que realizar a la vuelta de una llamada recursiva. La cuarta es la **linear recursion**, que es aquella en la que ninguna operación pendiente implica otra llamada recursiva. Por último, la quinta es la **tree recursion**, que ocurre cuando hay otra llamada recursiva en el conjunto de operaciones que se completarán después de que termine la recursión.
5. La recursión es un enfoque de arriba hacia debajo de la resolución de problemas. Divide el problema en pedazos o selecciona un paso clave, posponiendo el resto. Por otro lado, la iteración es más un enfoque de abajo hacia arriba. Ambos tienen sus limitaciones; por parte de los algoritmos recursivos encontramos que:

• Muchos lenguajes de programación no soportan recursión (por lo cual debe implementarse utilizando métodos iterativos).

• Aunque las funciones matemáticas se pueden implementar fácilmente utilizando la recursión, siempre es a costa de tiempo de ejecución adicional y espacio de memoria.

• Se puede llamar a una función recursiva desde dentro o fuera de sí misma, y para garantizar un funcionamiento adecuado, debe guardar las direcciones de retorno en algún orden para que el retorno a la ubicación adecuada produzca el resultado deseado cuando se realice el retorno a una instrucción de llamada.

Por otro lado, las limitaciones que presentan los métodos iterativos son las siguientes:

• El código iterativo no es legible y, por lo tanto, no es fácil de entender.

• En las técnicas iterativas, el bucle de las declaraciones es necesario y necesita una lógica compleja.

• Las iteraciones pueden dar lugar a un código largo.