Cálculo de la suma de una lista de números

Iniciamos nuestra investigación con un problema simple que usted ya sabe resolver sin recursividad. Suponga que usted desea calcular la suma de una lista de números como por ejemplo: [1,3,5,7,9][1,3,5,7,9]. Una función iterativa que calcula la suma se muestra en el [ActiveCode 1](https://runestone.academy/runestone/static/pythoned/Recursion/pythondsCalculoDeLaSumaDeUnaListaDeNumeros.html" \l "lst-itsum). La función utiliza una variable acumuladora (laSuma) para calcular el total acumulado de los números de la lista comenzando en 00 y sumando cada número de la lista.

RunLoad HistoryShow CodeLens

1

def sumalista(listaNumeros):

2

laSuma = 0

3

for i in listaNumeros:

4

laSuma = laSuma + i

5

return laSuma

6

​

7

print(sumalista([1,3,5,7,9]))

8

​

Sumatoria iterativa (lst\_itsum)

Imagine por un minuto que usted no tiene ciclos while o for. ¿Cómo calcularía la suma de una lista de números? Si usted fuera un matemático, podría comenzar recordando que la adición es una función que se define para dos parámetros, un pareja de números. Para redefinir el problema original de sumar una lista al problema alternativo de sumar parejas de números, podríamos reescribir la lista como una expresión completamente agrupada. Tal expresión tiene el siguiente aspecto:

((((1+3)+5)+7)+9)((((1+3)+5)+7)+9)

También podemos agrupar la expresión en el orden inverso,

(1+(3+(5+(7+9))))(1+(3+(5+(7+9))))

Observe que el conjunto más interno de paréntesis, (7+9)(7+9), es un problema que podemos resolver sin un ciclo o cualquiera otra instrucción especial. De hecho, podemos utilizar la siguiente secuencia de simplificaciones para calcular una suma final.

total= (1+(3+(5+(7+9))))total= (1+(3+(5+16)))total= (1+(3+21))total= (1+24)total= 25total= (1+(3+(5+(7+9))))total= (1+(3+(5+16)))total= (1+(3+21))total= (1+24)total= 25

¿Cómo podemos tomar esta idea y convertirla en un programa en Python? Primero, vamos a plantear el problema de la suma en términos de listas de Python. Podemos decir que la suma de la lista listaNumeros es la suma del primer elemento de la lista (listaNumeros[0]) y la suma de los números en el resto de la lista (listaNumeros[1:]). Para expresarlo en una forma funcional:

sumaLista(listaNumeros)=primero(listaNumeros)+sumaLista(resto(listaNumeros))sumaLista(listaNumeros)=primero(listaNumeros)+sumaLista(resto(listaNumeros))

En esta ecuación primero(listaNumeros)primero(listaNumeros) devuelve el primer elemento de la lista y resto(listaNumeros)resto(listaNumeros) devuelve una lista de todos los elementos menos el primero. Esto se expresa fácilmente en Python como se muestra en el [ActiveCode 2](https://runestone.academy/runestone/static/pythoned/Recursion/pythondsCalculoDeLaSumaDeUnaListaDeNumeros.html" \l "lst-recsum).

RunLoad HistoryShow CodeLens

1

def sumalista(listaNumeros):

2

if len(listaNumeros) == 1:

3

return listaNumeros[0]

4

else:

5

return listaNumeros[0] + sumalista(listaNumeros[1:])

6

​

7

print(sumalista([1,3,5,7,9]))

8

​

Sumatoria recursiva (lst\_recsum)

Hay algunas ideas clave que examinar en este programa. Primero, en la línea 2 estamos comprobando si la lista es de longitud uno. Esta comprobación es crucial y es nuestra cláusula de escape de la función. La suma de una lista de longitud 1 es trivial; es simplemente el número en la lista. Segundo, ¡en la línea 5 nuestra función se llama a sí misma! Ésta es la razón por la que llamamos recursivo al algoritmo sumalista. Una función recursiva es una función que se llama a sí misma.

La [Figura 1](https://runestone.academy/runestone/static/pythoned/Recursion/pythondsCalculoDeLaSumaDeUnaListaDeNumeros.html#fig-recsumin) muestra la serie de **llamadas recursivas** que se necesitan para sumar la lista [1,3,5,7,9][1,3,5,7,9]. Usted debe pensar en esta serie de llamadas como una serie de simplificaciones. Cada vez que hacemos una llamada recursiva estamos resolviendo un problema más pequeño, hasta llegar al punto en el que el problema no puede ser más pequeño.

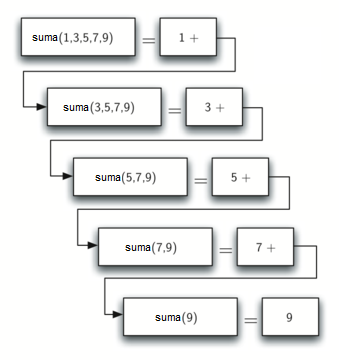


Figura 1: Serie de llamadas recursivas para sumar una lista de números

Figura 1: Serie de llamadas recursivas para sumar una lista de números

Cuando llegamos al punto en que el problema es tan simple como puede llegar a ser, comenzamos a juntar las soluciones de cada uno de los pequeños problemas hasta que el problema inicial se resuelva. La [Figura 2](https://runestone.academy/runestone/static/pythoned/Recursion/pythondsCalculoDeLaSumaDeUnaListaDeNumeros.html#fig-recsumout) muestra las sumas que se realizan a medida que sumalista funciona hacia atrás a través de la serie de llamadas. Cuando sumalista devuelve el resultado del problema superior, tenemos la solución de todo el problema.

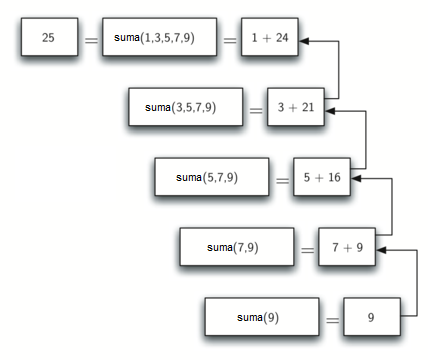


Figura 2: Series de devoluciones recursivas para sumar una lista de números

Figura 2: Series de devoluciones recursivas para sumar una lista de números