Tarea 1 - Análisis de Algoritmos

Cristian Ignacio Reyna Méndez

12 de Agosto del 2024

Nota: La longitud mínima requerida para ambos algoritmos es de al menos 1.

1. Algoritmo recursivo

Teorema: El algoritmo mult, el cual es recursivo, resuelve de manera correcta la multiplicación de todos los elementos de la lista.

Caso Base: Si la lista es de un único elemento entonces cae en el primer caso, de donde solamente va a retornar la primera instancia, es decir, el elemento colocado en el espacio "0"de la lista, y este será correcto como la multiplicación de todos los elementos de dicha lista.

Hip. Inductiva: Supongamos que n es igual a k y dicho algoritmo es válido para toda lista con k elementos.

Caso Inductivo:

P.D. Que el algoritmo es válido para toda k+1 elementos.

Supongamos una lista con k+1 elementos, de donde es importante ver que posee al menos 2 elementos, de esta forma, caerá en el segundo paso de donde multiplicaremos "la cabeza"de la lista por el algoritmo mult del resto de esta. Es decir, que multiplicaremos el k+1-esimo elemento por el algoritmo mult que recibe una lista con k-elementos, por H.I. el algoritmo mult es válido con k elementos, de esta manera, se multiplican el elemento k+1-esimo por todos los restantes elementos de la lista. Por lo tanto, el algoritmo mult devolverá correctamente la multiplicación de todos estos elementos.

1.1. Código solución - Recursivo

```
def mult(A):
    if(len(A) == 1):
        return A[0]
    else:
        return A[0] * mult(A[1:])
```

2. Algoritmo iterativo

Nota : El invariante en este caso para la demostración iterativa será que el acumulador dará el resultado correcto de multiplicar todos los k elementos de la lista.

Teorema: El algoritmo multi, el cual es iterativo, resuelve de manera correcta la multiplicación de todos los elementos de la lista.

Caso Base: Sea la lista de longitud 1, esta solamente se va a iterar 1 sola vez, de donde el acumulador con un valor de 1, se multiplicará con el único elemento v de la lista, así con el acc = acc * v; este v sera el resultado de la multiplicación de todos los elementos de la lista.

Hip. Inductiva: Tras ejecutar las lineas del for y acc *= v, un número de k veces, el invariante se mantiene.

Caso Inductivo:

P.D. Que el invariante se mantiene para una lista de k+1 elementos.

Sea una lista con k+1 elementos, el for se iterará k+1 veces de donde llamaremos v_i al i-ésimo elemento iterado. De este modo, en la 4ta línea se está iterando el acc = acc * v_i. Así, en la (i-1) vez, el acc valdrá... acc= v_0 * v_1 * ... * v_k (ya que por H.I. para k elementos, el invariante se mantiene) de este modo, en la i-ésima iteración, el acumulador operara... acc = v_0 * v_1 * ... * v_k * v_{k+1}.

Como el acumulador devuelve el resultado correcto para una lista de k elementos y ese acumulador se está multiplicando por v_{k+1} entonces el acumulador dará como resultado final la multiplicación de toda la lista de k+1 elementos, por lo tanto el invariante se mantiene.

2.1. Código solución - Iterativo

```
def multi(A):
    acc = 1
    for v in A:
        acc *= v
    return acc
```